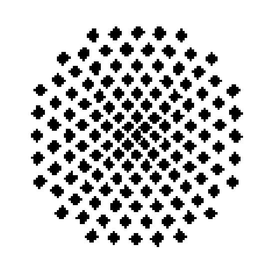


Universität Stuttgart
Institut für Automatisierungs- und Softwaretechnik
Prof. Dr.-Ing. Dr. h. c. P. Göhner

Zuverlässigkeit und Sicherheit von Automatisierungssystemen

Sommersemester 2015

Dr. N. Jazdi



www.ias.uni-stuttgart.de/zsa
zsa@ias.uni-stuttgart.de

Ansprechpartner für die Vorlesung

Bei organisatorischen Fragen zur Vorlesung oder bei Problemen mit dem Ablauf der Vorlesung „Zuverlässigkeit und Sicherheit von Automatisierungssystemen“ wenden Sie sich bitte an:

Dr.-Ing. Nasser Jazdi

Zimmer: 2.139 (Pfaffenwaldring 47, 2. Stock am IAS)

Tel.: 0711- 685-67303

E-Mail: zsa@ias.uni-stuttgart.de

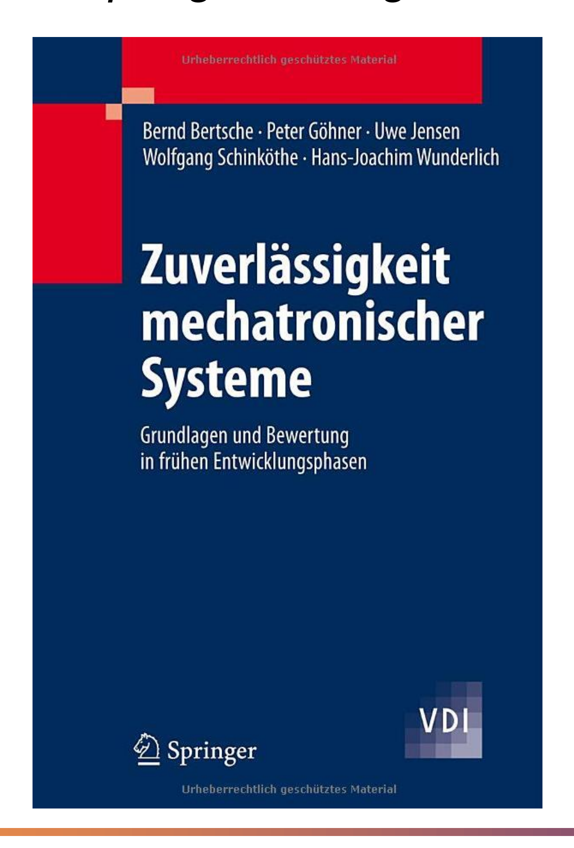
Unterlagen

- Skript „Zuverlässigkeit und Sicherheit von Automatisierungssystemen“
  - Blaue Texte zum Mitschreiben (im Skript nicht enthalten)
  - Live-Mitschriebe (leere Folien im Skript für Mitschrieb vorgesehen)
  - Fragen am Ende jedes Unterkapitels – Antworten zum Mitschreiben
- Vorlesungsportal im Internet: www.ias.uni-stuttgart.de/zsa
  - Aktuelle Informationen zur Vorlesung
  - Vollständige Vorlesungsunterlagen
  - Lecturnity-Aufzeichnungen im Internet und als Podcast
    - Vorlesungen

Literatur

Bertsche, Bernd; Göhner, Peter; Jensen, Uwe; Schinköthe, Wolfgang; Wunderlich, Hans-Joachim:

Zuverlässigkeit mechatronischer Systeme, Grundlagen und Bewertungen in frühen Entwicklungsphasen, *Springer-Verlag Berlin – Heidelberg*, 2009



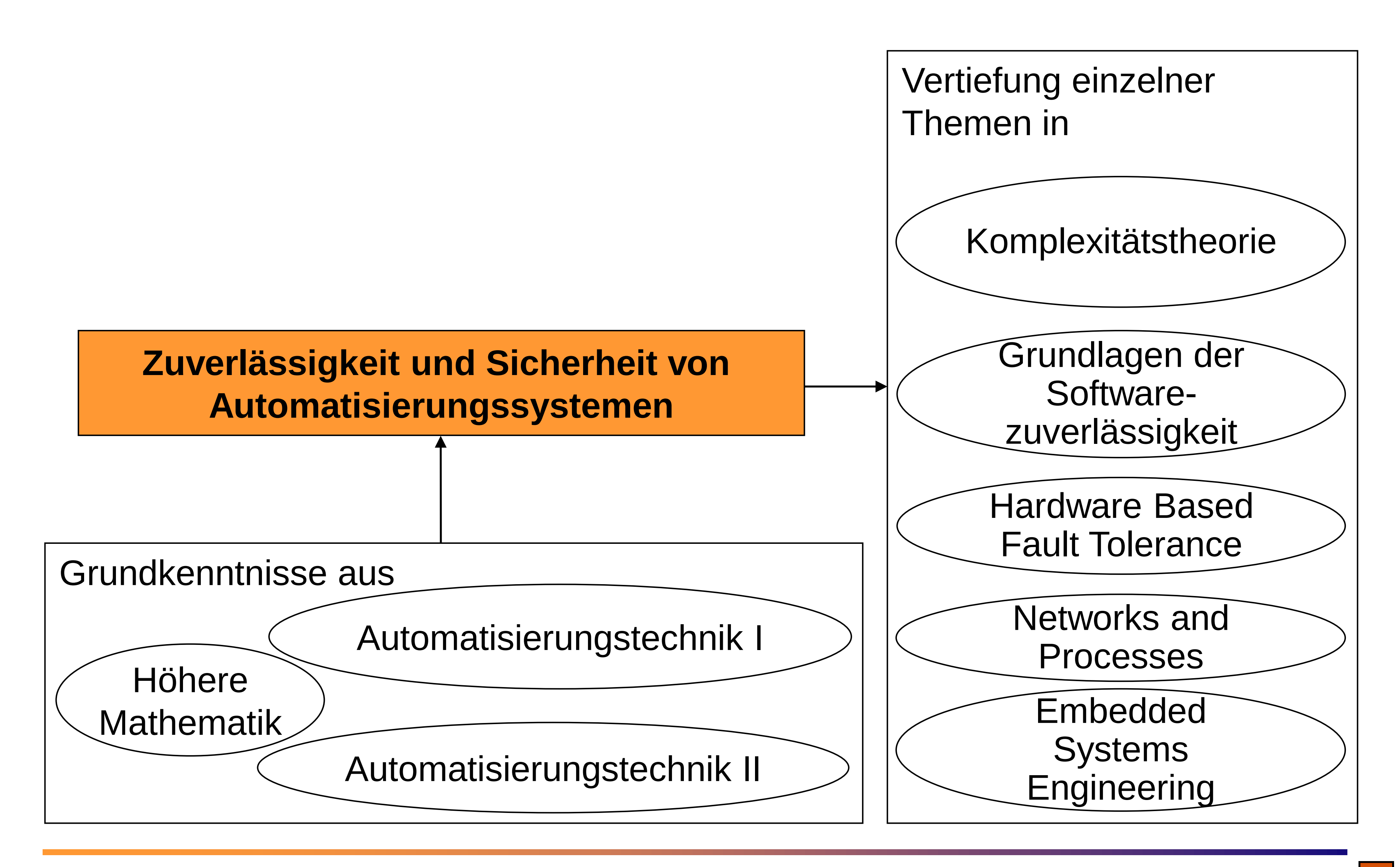
Vorlesungstermine

Nr.	Termin	Thema der Vorlesung
01	16.04.2015	Einführung in die Zuverlässigkeit und Sicherheit
02	23.04.2015	Begriffe und Normen
03	30.04.2015	Wahrscheinlichkeit und Lebensdauerverteilungen
04	07.05.2015	Verfügbarkeit und Zuverlässigkeitsberechnung
05	21.05.2015	Einführung in die Fehlerbaumanalyse
06	11.06.2015	Durchführung der Fehlerbaumanalyse
07	18.06.2015	Einführung in die Fehlermöglichkeits- und Einfluss-Analyse
08	25.06.2015	Durchführung der Fehlermöglichkeits- und Einfluss-Analyse
09	02.07.2015	Einführung in die Softwarezuverlässigkeit
10	09.07.2015	Modelle der Softwarezuverlässigkeit
11	16.07.2015	Zuverlässigkeits- und Sicherheitstechnik
12	23.07.2015	Übung / Prüfungsbesprechung

Ziele der Vorlesung

- Grundlagen der Zuverlässigkeit und Sicherheit verstehen
    - Kennenlernen der wichtigen Begriffe, Kenngrößen und Normen
    - Verstehen der Notwendigkeit von Zuverlässigkeits- und Sicherheitstechnik
- Zuverlässigkeitsmaßnahmen kennen und anwenden können
    - Methoden der Zuverlässigkeitsberechnung
    - Modelle der Zuverlässigkeitsanalyse
- Sicherheitskriterien berücksichtigen und handhaben
    - Strategien zum Entwurf sicherer Systeme
    - Bewertung von Risiko- und Gefährdungssituationen

Bezug zu anderen Vorlesungen der Fakultät:



Gliederung

§ 1 Einführung , Begriffe und Normen

1.1 Einführung in die Zuverlässigkeits- und Sicherheitstechnik

1.2 Definition von Zuverlässigkeit und Sicherheit

1.3 Wichtige Begriffe und Bedeutungen

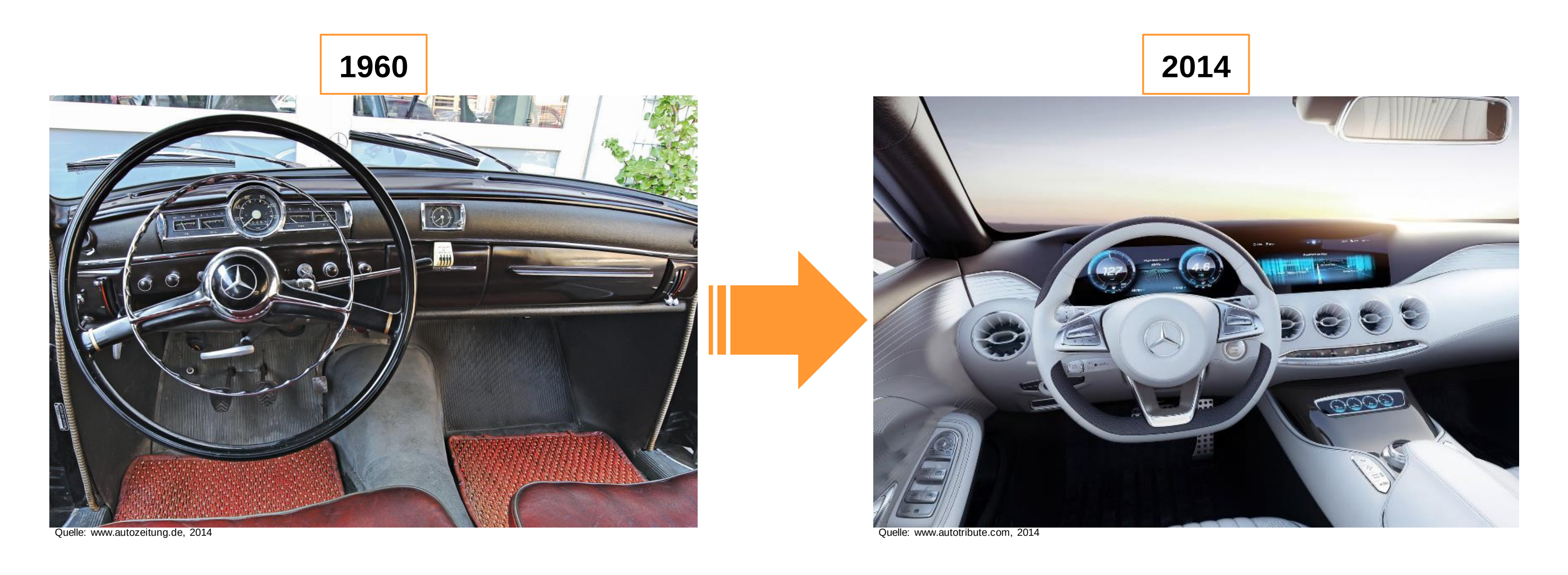
1.4 Normen

Wenn Systeme versagen...

- **Beispiel 1:** Bosch und Siemens Hausgeräte (BSH) in 2013
  - Rückruf von über 5 Millionen Geschirrspülmaschinen weltweit
  - Brandgefahr durch einen Fehler in einem elektr. Bauteil des Bedienfelds
  - Kunden erhalten kostenfreie Reparatur oder reduziertes Neugerät
  - Kosten für das Unternehmen ca. 700Mio Euro

- **Beispiel 2:** Toyota von 2010 bis 2013
  - Rückruf von über 22 Millionen Fahrzeugen in den USA
  - Ursachen u.a. klemmende Gaspedale, explodierende Airbags,…
  - Kosten der Rückrufaktionen für das Unternehmen nicht bekannt
  - Dazu aber mehrere Straf- bzw. Entschädigungs-Zahlungen an US-Kunden von insgesamt 2,6Mrd Dollar

Aktuelle Trends



- Systeme werden immer komplexer
- Menschen werden immer mehr von Systemen abhängig
- Fehler in Systemen haben Ausfälle oder Unfälle mit beträchtlichen Auswirkungen zur Folge:
  - Beschädigung von Güter
  - Verletzung und Tod von Menschen

Fazit

- Systeme müssen Zuverlässigkeit und Sicherheit gewährleisten
- Methoden der Zuverlässigkeits- und Sicherheitstechnik müssen beherrscht werden
- Nur dadurch kann weiterer Fortschritt gewährleistet werden
- Beispiel: Mercedes DICE *(Dynamic & Intuitive Control Experience)*



Quelle: www.daimler.com, 2014

Historie (1940er – 1980er)

- Bis 1940
  - Festlegung von produktspezifischen Qualitäts-Merkmalen
  - Einführung von Qualitäts-Prüfungen
- 1940 – 1960 (mit wachsender Zahl von Systemen)
  - Systematische Erfassung und Analyse von Ausfällen
  - Einführung statistischer Qualitätskontrollen
  - Zuverlässigkeitsuntersuchungen bereits während der Entwicklung
- 1960 – 1980 (mit wachsender Komplexität der Systeme)
  - Methoden zur Schätzung und zum Nachweis von Ausfallzeiten
  - Festlegung von Zuverlässigkeitszielen bereits während der Analyse
  - Zunehmende Bedeutung der Softwarezuverlässigkeit

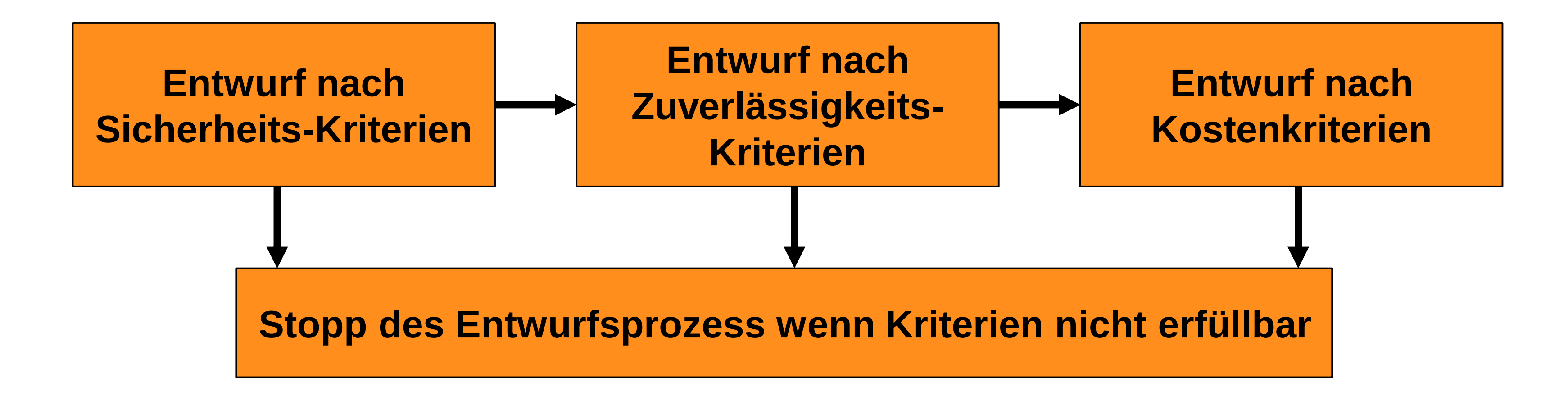
Historie (1980er – heute)

- 1980 – 1990
  - Detailliertere Untersuchung der Software-Zuverlässigkeit
  - Notwendigkeit automatisierter Tests bzw. Selbsttests von Systemen
  - Steigende Bedeutung der Produkthaftung

- 1990 bis heute
  - Gegenseitiger Austausch auf nationaler und internationaler Ebene
  - Erarbeitung einheitlicher Richtlinien (nicht produktspezifisch!)
  - Qualitätsmerkmale als Grundsätze bzw. Ziele der Entwicklung
  - Einbeziehung von Kunden und Lieferanten

Bedeutung im Entwicklungsprozess

- Kundenzufriedenheit ist elementarer Aspekt
- Komplexer werdende Systeme sind teurer als ihre Vorgänger
- Erwartungen der Kunden an Zuverlässigkeit und Sicherheit der Systeme steigen mit dem Preis
- Aspekte der Zuverlässigkeit und Sicherheit müssen im Entwicklungsprozess bereits in frühen Phasen berücksichtigt und besonders beachtet werden



Zuverlässigkeitsmanagement

- Organisation der Zuverlässigkeitsarbeit im Unternehmen (nach VDI 4003)
- Gesamtheit von Planung, Durchführung und Kontrolle der Zuverlässigkeit
- Dazu gehören die Ziele:
  - Nachweis einer definierten Ausfallwahrscheinlichkeit eines Produkts bzw. Systems
  - Optimierung der Verfügbarkeit über den geplanten Lebenszyklus
  - Verbesserungen durch Vergleiche mit alternativen Systementwürfen
  - Identifikation kritischer Komponenten oder Teilsysteme
  - Gewinnung von Planungswerten
  - Definition allgemeingültiger Zuverlässigkeitsziele
  - Aufbau einer Wissensbasis

Beispiel Daimler AG: „Sieben kategorische Imperative"

- Die Zuverlässigkeit elektronischer Systeme muss mindestens genauso groß sein wie die vergleichbarer mechanischer Systeme!

- Wir brauchen Standards!
- Softwarefehler sind kein Zufall!

- Software-Entwicklungs-Tools dürfen nicht schnellere Fortschritte machen als Software-Validierungs-Tools!
- Zertifizierung und Systemintegration sind Domänen der Erstausrüster (OEM)!

- Keine Hardware-Fehler!
- Keine Funktionen, die dem Kunden keinen Nutzen bieten!

§ 1 Einführung , Begriffe und Normen

Zuverlässigkeit

– **Wahrscheinlichkeit**, dass ein System

1. innerhalb eines bestimmten Zeitintervalls,
2. unter den zulässigen Betriebsbedingungen,

die geforderte, spezifizierte Funktion erfüllt.

– Maßnahmen der Zuverlässigkeitstechnik richten sich gegen das Auftreten von Fehlern und Ausfällen

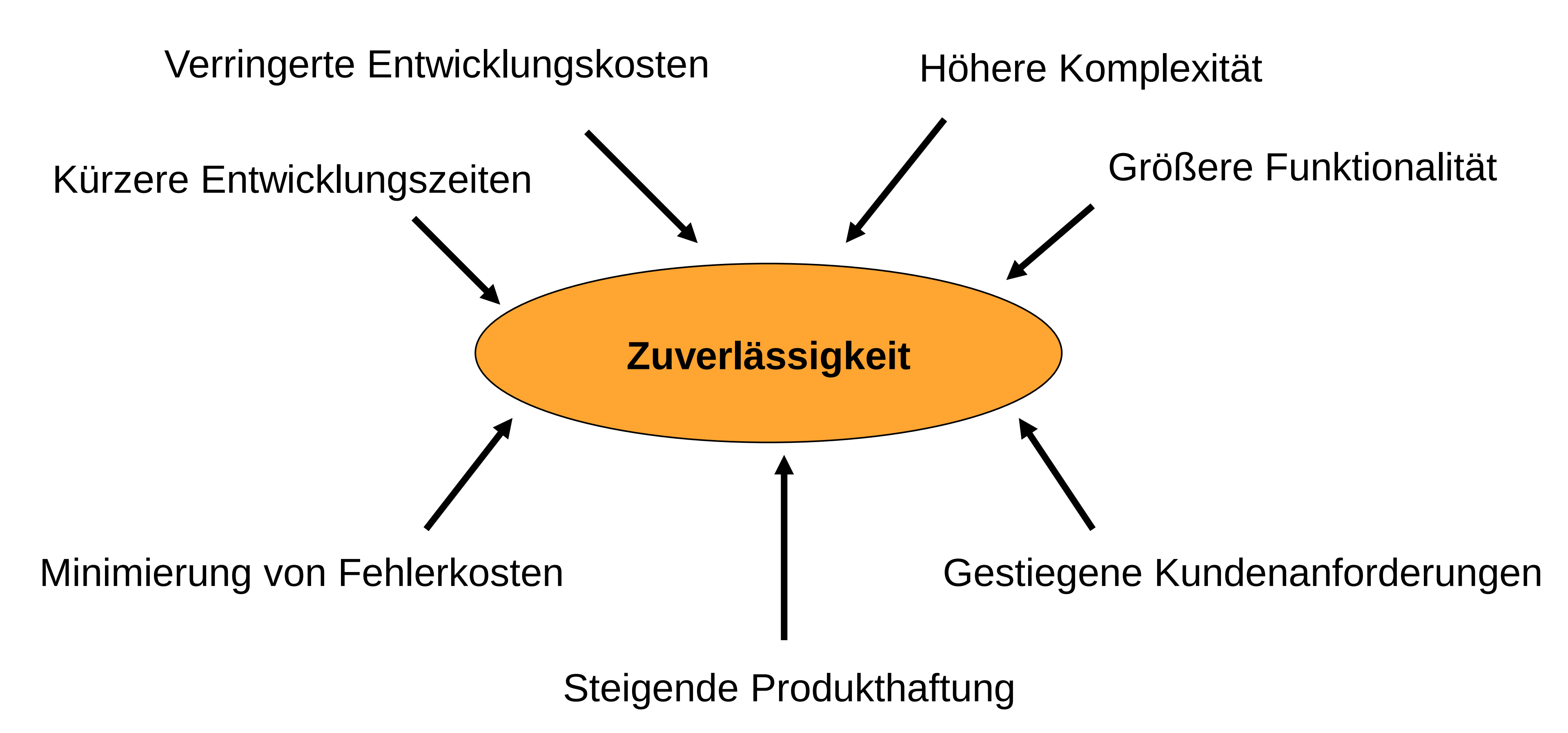
– Grund für Maßnahmen ist die Wirtschaftlichkeit

– Nachweisverfahren: Zuverlässigkeitsanalyse

– Ziel ist die Erfüllung der Garantiezeiten

– Zuverlässigkeit ist elementarer Teil der Qualität (Qualitätsmanagement)

Einflussfaktoren auf die Zuverlässigkeit



Zuverlässigkeitsanalyse

- Ziele:
  - Prognose der zu erwartenden Zuverlässigkeit
  - Erkennung und Beseitigung von Schwachstellen
  - Durchführung von Vergleichsstudien

- Anforderungen an die Durchführung:
  - Erfahrenes Bearbeitungsteam
  - Systematische Planung aller erforderlichen Arbeitsschritte
  - Geeignete Zuverlässigkeitsdatenbasis
  - Geeignete Software

Quantitative Modelle der Zuverlässigkeitsanalyse

- Probabilistische Zuverlässigkeitsprognose
- Berechnung einer vorausgesagten und erwarteten Zuverlässigkeit
- Analyse der Ausfallrate

- Beispiel:
  - Fehlerbaumanalyse FTA (vgl. Kap 3)
  - Lebensdauerverteilungen (vgl. Kap 2)
  - Boole-Modell (vgl. Kap 2)
  - Markov-Modell

Markov-Modell (1/2)

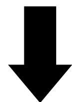

Quelle: de.wikipedia.org, 2014

- Benannt nach russischem Mathematiker A. A. Markov
- Stochastisches Modell, bei dem ein System über unbeobachtete Zustände modelliert wird
- Übergange zwischen den Zuständen und das Verweilen in einem Zustand werden mit Wahrscheinlichkeiten versehen
- Beispiel: Beschreibung des Systemverhaltens über ein Zustandsdiagramm mit den zwei Zuständen 1 (System ist in Ordnung) und 2 (System ist ausgefallen)

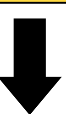
Markov-Modell (2/2)

– Ablauf:

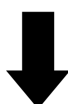
System



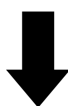
Definition von Zuständen und Übergängen



Zusammenstellung und Beschreibung der statistischen Eingangsdaten durch Differentialgleichungen (DGL)



Numerische Auswertung der DGL



Zuverlässigkeitskenngrößen

Qualitative Modelle der Zuverlässigkeitsanalyse

- Systematische Untersuchung der Auswirkungen von Fehlern oder Ausfällen
- Aufdeckung von Schwachstellen in Systemen
- Analyse der Ausfallart

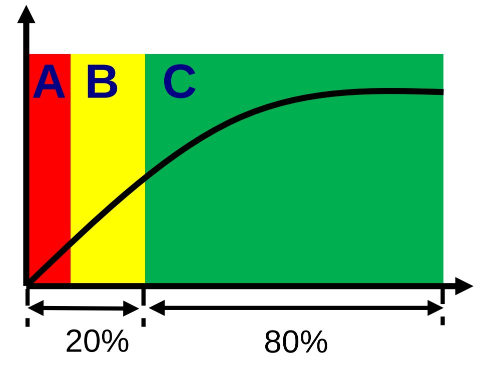
- Beispiele:
  - Fehlerbaumanalyse FTA (vgl. Kap 3)
  - Fehlermöglichkeits- und Einfluss-Analyse FMEA (vgl. Kap 4)
  - Checklisten (vgl. Kap 4)
  - ABC-Analyse

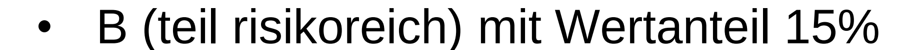
ABC-Analyse (1/2)

- Beschreibung von H. Ford Dickie (Manager bei General Electric) in 1951
- Ziel ist nach einer Zustandsbeschreibung die Trennung des „Wesentlichen“ vom „Unwesentlichen“ und dadurch die Steigerung der Wirtschaftlichkeit
- Basiert auf:
  - Pareto Prinzip (nach Vilfredo Pareto, 1896)
    - 80% der Ergebnisse eines Projekts in 20% der Zeit
    - 20% der Ergebnisse eines Projekts in 80% der Zeit
  - Lorenz-Kurve (nach Max Otto Lorenz, 1905)
    - Statistische Verteilungskurve
    - Berücksichtigt das Ausmaß bzw. den Grad der Ungleichverteilung einer Größe zu einer anderen Größe

ABC-Analyse (2/2)

- Vorhandene Daten werden in drei Relevanz-Klassen eingeteilt und nach Größe und Wichtigkeit geordnet und visualisiert



- Beispiel Beanspruchung eines Bauteils:
  - A (risikoreich) mit Wertanteil 5%
    - Beanspruchung z.B. statische Belastungen
    - Ausfallverhalten über Berechnung bestimmbar
  - 
    B (teil risikoreich) mit Wertanteil 15%
    - Beanspruchung z.B. Verschleiß
    - Ausfallverhalten über Schätzung oder Experimente bestimmbar
  - C (risikoneutral) mit Wertanteil 80%
    - Beanspruchung z.B. stochastische Stöße
    - Ausfallverhalten kaum bzw. nicht bestimmbar

Induktive Zuverlässigkeitsanalyse

- Es wird ein Schluss vom Einzelnen auf die Allgemeinheit gezogen
- Ziel ist die Ableitung von allgemeinen Regeln oder Zusammenhängen
- Vereinfachter Gedankengang:

Mein Smartphone benötigt Strom → **Mein Smartphone ist ein elektrisches Gerät** → **Alle elektrischen Geräte benötigen Strom**

In der Zuverlässigkeitstechnik:

- Vorwärts gerichtete Verfolgung von Ereignissen, die zu Unfällen führen können
- Frage: „Was sind die Folgen für das Gesamtsystem, wenn das Einzelereignis X eintritt“?

Deduktive Zuverlässigkeitsanalyse

- Es wird ein Schluss von der Allgemeinheit auf das Einzelne gezogen
- Ziel ist die Schlussfolgerung logischer Konsequenzen
- Vereinfachter Gedankengang:

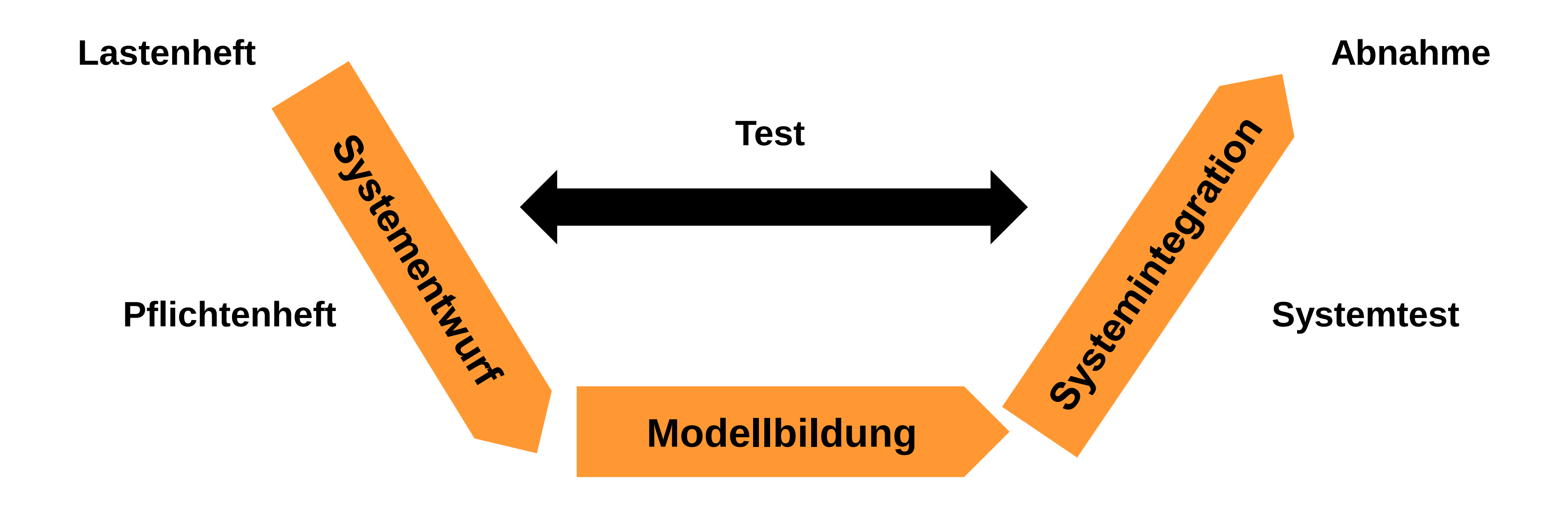
Elektrische Geräte benötigen Strom → **Mein Smartphone ist ein elektrisches Gerät** → **Auch mein Smartphone benötigt Strom**

In der Zuverlässigkeitstechnik:

- Rückwärts gerichtete Herleitung von möglichen Ausfällen oder Fehlerquellen, die zu Unfällen führen können
- Frage: „Welche Einzelereignisse oder Kombinationen von Einzelereignissen können zu einem unerwünschten Zustand führen?“

Zuverlässigkeitsanalyse in Entwicklungsprozess

– Vereinfacht dargestellt am Beispiel des V-Modells:



Sicherheit

- **Sachlage**, dass von einem System

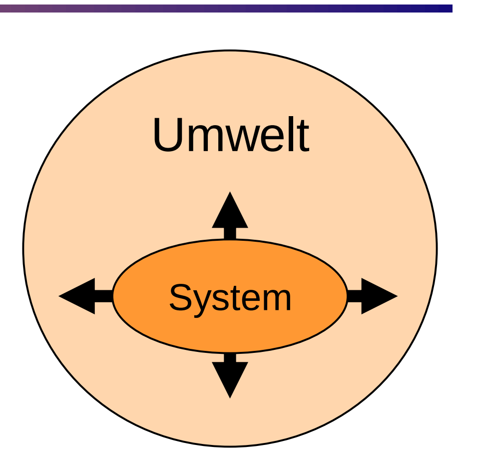
    1. innerhalb vorgegebenen Grenzen und

    2. für eine bestimmte Zeitdauer

  keine Gefahr ausgeht oder keine Gefährdung eintritt.

- Maßnahmen der Sicherheitstechnik richten sich gegen gefährliche Auswirkungen von Fehlern und Ausfällen
- Grund für Maßnahmen ist die Genehmigung durch eine Zulassungsbehörde
- Nachweisverfahren: Sicherheitsnachweis über Gefahrenanalyse
- Ziel ist das Verhindern einer Gefahr
- Sicher sein bedeutet frei sein von Gefährdungen

Sicherheit = Safety

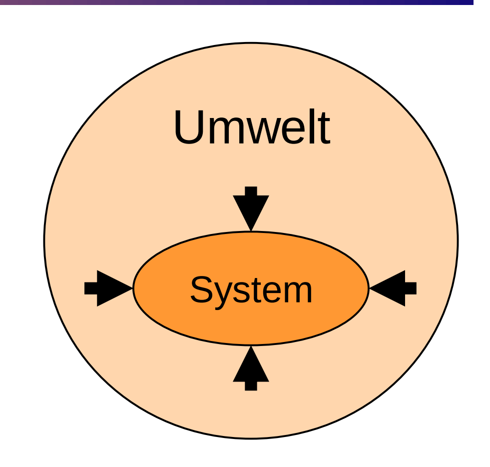


- Verhindern von Gefahren, die vom System auf die Umwelt oder den Mensch ausgehen
- Gefahr wirkt von Innen nach Außen
- Ziel: Erfüllung von (rechtlichen) Sicherheitsstandards



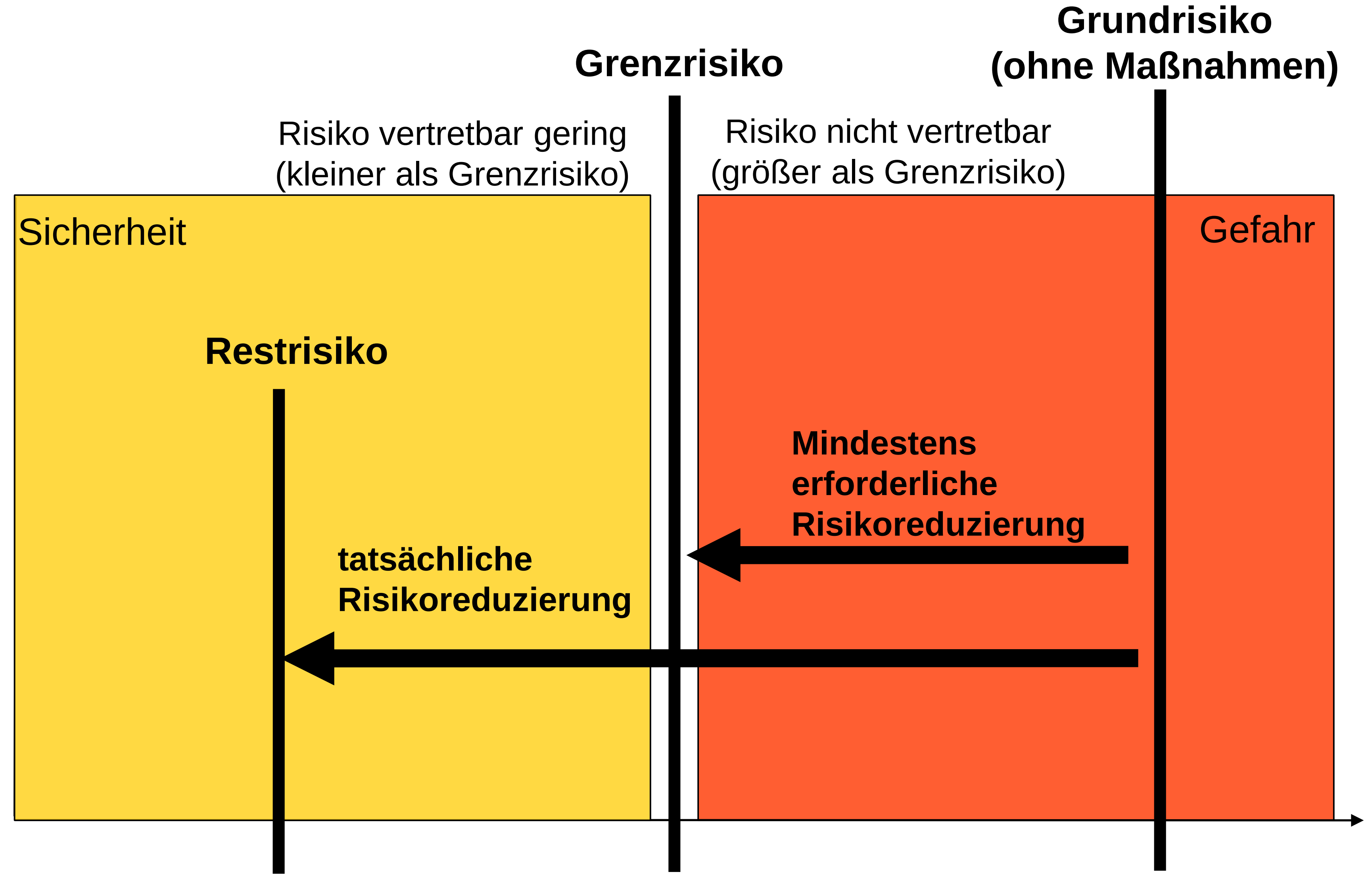
- Sicherheitsmanagement während der Entwicklung (nach ISO 26262):
  - Sicherheitsplan (*safety plan*)
    - Planung von Entwicklungsaktivitäten und Analysemethoden
    - Ableitung von Verifikationsmaßnahmen
  - Sicherheitsnachweis (*safety case*)
    - Ergebnisse der Aktivitäten und Maßnahmen
    - Nachweis, dass die sicherheitskritischen Anforderungen vollständig erfüllt werden

Sicherheit = Security



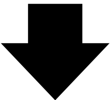
- Verhindern von Gefahren, die vom Menschen oder der Umwelt auf das System einwirken
- Gefahr wirkt von Außen nach Innen
- Beinhaltet Außenzugriffe (Hackerangriffe, Viren,…), die das System so beeinflussen können, dass es gefährlich werden kann
- Ziel: Schutz vor Bedrohungen und Vermeidung wirtschaftlicher Schäden
- Informationstechnik beinhaltet die Schutzziele:
  - Vertraulichkeit (*confidentiality*): Zugriff nur für autorisierte Benutzer
  - Integrität (*integrity*): Verhinderung unbemerkter Veränderungen
  - Verfügbarkeit (*availability*): Verhinderung von Systemausfällen
  - Authenzität (authenticity): Echtheit und Überprüfbarkeit von Daten

Risikobetrachtung

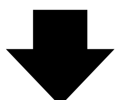


Sicherheitsnachweis

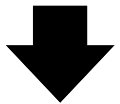
Beschreibung des Automatisierungssystems



Beschreibung der Umgebung und Einflüsse



Gefahrenanalyse
(Identifikation von Gefährdungen)



Risikobewertung
(Bewertung der Auswirkungen von Gefährdungen)

Frage zu Kapitel 1.2

Welchen Aussagen stimmen Sie zu?

- ☐ Eine Anlage mit einer hohen Sicherheit zeichnet sich grundsätzlich auch durch eine hohe Zuverlässigkeit aus.
- ☐ Grund für Zuverlässigkeitsmaßnahmen von Systemen ist die Wirtschaftlichkeit.
- ☐ Absolute Sicherheit ohne jegliches Risiko gibt es in der Technik nicht.
- ☐ Die Berechnung der Zuverlässigkeit ist das Ziel einer qualitativen Analyse.

§ 1 Einführung , Begriffe und Normen

1.1 Einführung in die Zuverlässigkeits- und Sicherheitstechnik

1.2 Definition von Zuverlässigkeit und Sicherheit

1.3 Wichtige Begriffe und Bedeutungen

1.4 Normen

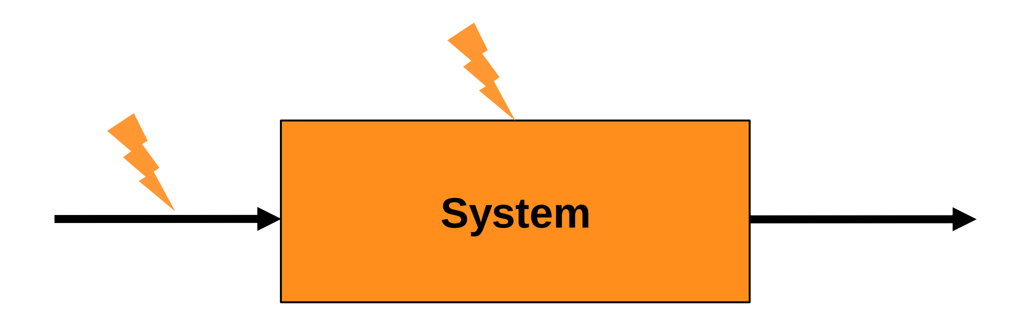
Begriff: RAMS(S)-Management nach DIN EN 50126

- Prozess bzw. Methodik zur Verhinderung von Fehlern bereits in der Planungsphase
- Dient zur Systemspezifikation von sicherheitskritischen Systemen
- RAMS(S)-Prozess ist erst abgeschlossen, wenn System bzw. Produkt außer Betrieb genommen und entsorgt worden ist
- Beinhaltet:
  - ***R**eliability* (Zuverlässigkeit)
  - ***A**vailability* (Verfügbarkeit)
  - ***M**aintainability* (Instandhaltbarkeit, Dauer von Reparaturen)
  - ***S**afety* (Sicherheit, Gefährdungsfreiheit des Gerätes nach außen)
  - ***S**ecurity* (Sicherheit, Schutz gegen unberechtigten Zugriff)

Begriff: Fehler

- Fehler
  - Nichterfüllung mindestens einer vorgegebenen Forderung
  - Fehler entstehen durch Mensch, Material, Milieu (Umgebung), Methode,…

- Im engl. klare Unterscheidung (nach ISO 26262):
  - *fault*: „Eine abnormale Bedingung, die ein Element scheitern lassen kann."
  - *error*: „Abweichung zwischen dem berechneten und dem spezifizierten Wert."



Arten von Fehlern (1/2)

- Physikalische Fehler
  - Meist stochastischen Ursprungs
  - Ursachen sind physikalische oder chemische Ausfallmechanismen oder Effekte (z.B. elektromagnetische Störungen)
  - Charakterisiert durch konstante Ausfallraten:

- Inhärente Fehler
  - Fehler, die schon vor Beginn des Betriebs vorhanden sind
  - Oft systematische Fehler, die nicht immer offensichtlich sind:
    - Pflichtenheftfehler
    - Software-Entwurfsfehler
    - Verdrahtungsfehler

Arten von Fehlern (2/2)

- Nicht inhärente Fehler
  - Fehler, die erst nach der Inbetriebnahme begangen werden
  - Oft kaum abzusehen bzw. zu verhindern:
    - Bedienfehler
    - Fehler bei Wartung und Pflege
    - Absichtliche Fehler (Sabotage oder Vandalismus)

Fehlertypen / Fehlerübergänge

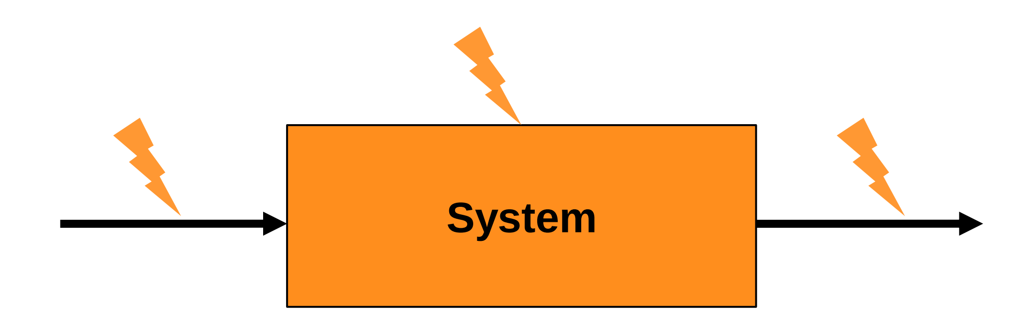
- Statischer Fehler
  - Übergang in Fehlerzustand statisch, d.h., Komponente wird abrupt defekt und bleibt defekt bis zur Reparatur
  - Dient als Annahme für einfache Zuverlässigkeitsmodelle
- Drift-Fehler
  - Langsamer Übergang der Systemmerkmale in einen Fehlerzustand
  - Meist bei analogen, weniger bei digitalen Bauteilen
- Transienter Fehler
  - Kurze Wirkungsdauer bedingt durch externe Einstreuung (z.B. EMV)
  - Oft vernachlässigbar wenn Wirkungsdauer sehr gering
- Intermittierender Fehler
  - Häufig auftretender transienter Fehler mit nachfolgender Ruhepause
  - Oft durch Betrieb der Hardware an Belastungsgrenze

Umgang mit Fehlern - Maßnahmen zur....

- Fehlerprävention (*fault prevention*)
  - Um das Auftreten von Fehlern von vornherein zu vermeiden.
- Fehlertoleranz (*fault tolerance*)
  - Um auch mit begrenzter Anzahl von fehlerhaften Komponenten noch die geforderte Funktion erfüllen zu können.
- Fehlerbehebung (*fault removal*)
  - Um die Anzahl oder Auswirkungen von Fehlern zu vermindern.
- Fehlervorhersage (*fault forecasting*)
  - Um die zukünftige Zahl von Fehlern und deren Konsequenzen abschätzen zu können.

Begriff: Ausfall

- Ausfall (*failure*)
  - Aussetzen der Ausführung einer festgelegten Aufgabe
  - Übergang vom funktionsfähigen in den fehlerhaften Zustand
- Unterscheidung:
  - *hard failure:* Sofortiger Übergang zum völligen Versagen des Systems
  - *soft failure:* Übergang hin zu einer unzulässigen Abweichung (Drift)



Begriffe: Sicherer Zustand und Rückfallebene

- Sicherer Zustand
  - Zustand, in dem trotz gewisser, zugelassener Ausfälle keine Gefahr mehr vom bzw. auf das System ausgehen kann
  - Beispiel: Haltezustand von Zügen

- Rückfallebene
  - Sekundärsystem, das bei Ausfall des primären Systems einen Schutz gegenüber einer Gefährdung bietet und den Totalausfall des Systems verhindert
  - Meist Aufrechterhaltung einer reduzierten Betriebsqualität

Übergange in einen sicheren Zustand

- *fail-operational*
  - Grundfunktionalität wird solange aufrechterhalten, bis es möglich ist, einen sicheren Zustand zu erreichen (auch über längere Dauer hinweg)
  - System beinhaltet redundante Strukturen um den Betrieb weiterzuführen
- *fail-safe*
  - Unmittelbarer Übergang in einen sicheren Zustand
  - System bleibt bis zur Reparatur im sicheren Zustand
- *fail-silent*
  - Sofortiges Stillhalten oder Abschalten von Teilsystemen oder Komponenten nach Detektion einer Abweichung vom vorgegebenen Betrieb
  - Ziel ist es, die Beeinflussung anderer Teilsystemen zu verhindern

Begriffe: Verfügbarkeit und Verlässlichkeit

- Verfügbarkeit
  - Wahrscheinlichkeit, ein reparierbares System zu einem vorgegebenen Zeitpunkt im funktionsfähigen Zustand anzutreffen
  - Wesentliche Kennzahl eines Systems

- Verlässlichkeit
  - Eigenschaft eines Systems die es erlaubt, volles Vertrauen in die bereitgestellte Funktionalität zu setzen
  - Überbegriff für Qualitätskriterien eines Systems

Übersicht Begriffe der Zuverlässigkeits-/Sicherheitstechnik

Begriff	Definition
Fehler$_1$ (Error)	Abweichung zwischen einem berechneten Wert und dem wahren, spezifizierten oder theoretisch richtigem Wert aufgrund eines Fehlers$_2$ oder einer Störung$_1$.
Fehler$_2$ (Defect)	Nichterfüllung der Anforderungsspezifikation, Unkorrektheit.
Fehler$_3$ (Fault)	Abweichung der tatsächlichen Ausführung von der für die Erfüllung der Anforderungsspezifikation erforderlichen konstruktiven und fertigungstechnischen Ausführung des Systems.
Fehler$_4$ (Mistake)	Menschliche Handlung mit unerwünschtem Ergebnis, ein Irrtum oder Schnitzer.
Gefahr	Sachlage, bei der das Risiko größer als das Grenzrisiko ist, wobei unter Grenzrisiko das größte noch vertretbare Risiko verstanden wird.
Korrektheit	Erfüllung der Anforderungsspezifikation. Übereinstimmung zwischen realisierter und spezifizierter Funktion.
Risiko	Möglichkeit, Schaden zu erleiden. Risiko = Schadensausmaß $*$ Wahrscheinlichkeit des Schadeneintritts

Übersicht Begriffe der Zuverlässigkeits-/Sicherheitstechnik

Begriff	Definition
Schaden	Nachteil durch Verletzung von Rechtsgütern aufgrund eines bestimmten technischen Vorgangs oder Zustandes.
Schutz	Verringerung des Risikos durch Maßnahmen, die entweder die Eintrittshäufigkeit oder das Ausmaß des Schadens oder beide einschränken.
Sicherheit	Sachlage, bei der das Risiko kleiner als das Grenzrisiko ist.
Störung$_1$ (Interference)	Vorübergehende Beeinträchtigung einer Funktion.
Störung$_2$ (Deficiency)	Fehlende, fehlerhafte oder unvollständige Erfüllung einer geforderten Funktion.
Validation	Nachweis, dass ein System seinen Erfordernissen genügt.
Verifikation	Nachweis, dass eine Betrachtungseinheit die Anforderungsspezifikation vollständig erfüllt.
Versagen	Verhalten eines Systems, das nicht der Anforderungsspezifikation entspricht.
Robustheit	Fähigkeit einer Betrachtungseinheit, auch bei Verletzung der spezifizierten Randbedingungen vereinbarte Funktionen zu erfüllen.

Frage zu Kapitel 1.3

Welchen Aussagen stimmen Sie zu?

- ☐ Ein Fehler ist ein Zustand.
- ☐ Ein Fehler hat immer einen Ausfall zur Folge.
- ☐ Verifikation ist der Nachweis, dass ein System den Erfordernissen genügt.
- ☐ Verfügbarkeit steht für Sicherheit + Zuverlässigkeit + Verlässlichkeit.

§ 1 Einführung , Begriffe und Normen

1.1 Einführung in die Zuverlässigkeits- und Sicherheitstechnik

1.2 Definition von Zuverlässigkeit und Sicherheit

1.3 Wichtige Begriffe und Bedeutungen

1.4 Normen

Allgemeines zu Normen

- Norm: Regel, Maßstab, Richtung,…
- Normung: Formulierung, Veröffentlichung und Anwendung von Regeln, Leitlinien und Merkmalen durch eine anerkannte Organisation
- Nationale Normung:
  - Deutsches Institut für Normung (DIN)
  - Verein Deutscher Ingenieure (VDI)
- EU-Normung:
  - Europäisches Komitee für Normung (CEN)
  - Europäisches Komitee für elektrotechnische Normung (CENLEC)
- Internationale Normung:
  - Internationale Organisation für Normung (ISO)
  - Institut of Electrical and Electronics Engineers (IEEE)

Nationale Normungsarbeit nach DIN

- Historie
  - 1917 Gründung „Normenausschuss der deutschen Industrie“
  - 1926 Umbenennung zu „Deutscher Normenausschuss“ und Gründung von zusätzlichen Arbeitsbereichen für Krankenhäuser, Bürowesen etc.
  - 1975 Zusammenfassung aller Arbeitsbereiche zum „Deutschen Institut für Normung“ (DIN) und Gründung von Normenausschüsse, z.B.:
    - Normenausschuss Automobiltechnik (NAAutomobil)
    - Normenstelle Elektrotechnik (NE)
- DIN hat ca.1800 Mitglieder aus Unternehmen, Behörden oder Wissenschaft
- Aufgabe ist die Erarbeitung von Normen innerhalb “interessierter Kreise“, die sich aus Vertretern der verschiedenen Normenausschüsse bilden
- Jeder kann einen Antrag auf Normung stellen

Umgang mit Normen

- Normen sind kein Gesetz, d.h., deren Einhaltung ist nicht verbindlich!
- Normen repräsentieren den zu diesem Zeitpunkt aktuellen Stand der Technik
- Stand der Technik entspricht dem, was heute technisch machbar ist und tatsächlich mindestens einmal realisiert wurde
- Für neue sicherheitskritische Systeme:

Achtung:

- Im Produkthaftungsfall (nicht Schäden am Produkt, sondern Schäden durch das Produkt) ist <u>nicht</u> der Stand der Technik, sondern der Stand von Wissenschaft und Technik nachzuweisen
- Stand von Wissenschaft und Technik entspricht dem, was heute erkenntnistheoretisch erreichbar ist (höchstes Niveau)
- Für neue sicherheitskritische Systeme:

Chancen und Vorteile von Normen für ein Unternehmen

- Kostensenkungen in Entwicklung und Produktion
  - Massenproduktion
  - Globaler Einkauf
  - Verminderte Anpassungskosten
  - Verkürzung der Entwicklungszeiten
- Wettbewerbsvorteile gegenüber Konkurrenten
  - Wissens- und Zeitvorteile
  - Imagesteigerung
  - Zugang zu neuen Märkten
- Klar definierte Schnittstellen sowohl intern als auch zu Kunden und Zulieferern
- Verbesserung der Kundenkontakte und Anreize für neue Kunden

Sicherheitsgrundnorm DIN EN 61508

- Funktionale Sicherheit sicherheitsbezogener elektrischer / elektronischer / programmierbar elektronischer Systeme
- Aufbau (Stand: 02/2011):
  - **Teil 0:** Funktionale Sicherheit
  - **Teil 1:** Allgemeine Anforderungen
  - **Teil 2:** Anforderungen an sicherheitsbezogene elektrische, elektronischer und programmierbar elektronischer Systeme
  - **Teil 3:** Anforderungen an Software
  - **Teil 4:** Begriffe und Abkürzungen
  - **Teil 5:** Beispiele zur Ermittlung der Stufe der Sicherheitsintegrität (SIL)
  - **Teil 6:** Anwendungsrichtlinie für IEC 61508-2 und IEC 61508-3
  - **Teil 7:** Überblick über Verfahren und Maßnahmen

Ausschnitt DIN EN 61508 (1/3)

Funktionale Sicherheit elektrischer/elektronischer/programmierbar elektronischer sicherheitsbezogener Systeme

Teil 1: Allgemeine Anforderungen

1 Anwendungsbereich

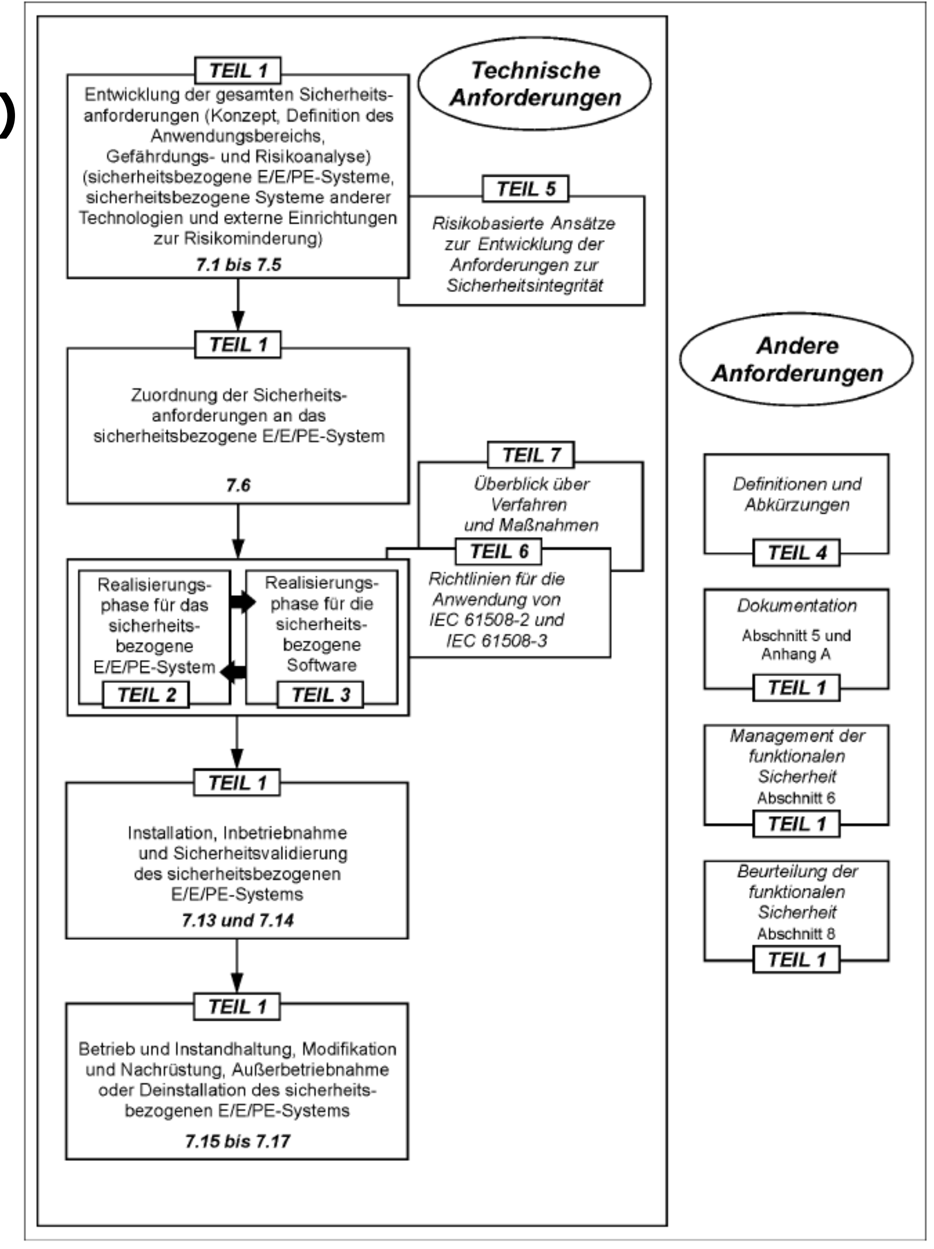
1.1 Diese Internationale Norm behandelt diejenigen Gesichtspunkte, die zu betrachten sind, wenn elektrische/elektronische/programmierbar elektronische Systeme (E/E/PES) zur Ausführung von Sicherheitsfunktionen eingesetzt werden. Ein Hauptziel dieser Norm ist es, für ein bestimmtes Anwendungsgebiet die Entwicklung einer entsprechenden Internationalen Norm durch das jeweils verantwortliche Komitee zu ermöglichen. Dies wird es erlauben, alle wichtigen Einflussgrößen dieses Anwendungsgebietes vollständig zu berücksichtigen und damit dessen besonderen Erfordernissen nachzukommen. Ein zweites Ziel dieser Norm ist es, die Entwicklung eines elektrischen/ elektronischen/programmierbar elektronischen (E/E/PE) sicherheitsbezogenen Systems, für dessen Anwendungsgebiet noch keine Internationale Norm besteht, zu ermöglichen.

1.2 Insbesondere:

a) gilt diese Norm für sicherheitsbezogene Systeme, wenn eines oder mehrere dieser Systeme elektrische/elektronische/programmierbar elektronische Geräte enthalten;

ANMERKUNG 1 Für einfache sicherheitsbezogene E/E/PE-Systeme geringer Komplexität können bestimmte, in dieser Norm festgelegte Anforderungen unnötig sein, und eine Befreiung von der Normerfüllung in Bezug auf solche Anforderungen ist möglich (siehe 4.2 und Definition eines einfachen sicherheitsbezogenen E/E/PE-Systems in 3.4.4 von IEC 61508-4).

Ausschnitt DIN EN 61508 (2/3)



Ausschnitt DIN EN 61508 (3/3)

8 Beurteilung der funktionalen Sicherheit

8.1 Ziel

Das Ziel der Anforderungen dieses Abschnitts ist es zu untersuchen und zu beurteilen, ob die funktionale Sicherheit durch die sicherheitsbezogenen E/E/PE-Systeme erreicht worden ist.

8.2 Anforderungen

8.2.1 Es müssen eine oder mehrere Personen für die Durchführung der Beurteilung der funktionalen Sicherheit ernannt werden, um zu einer Beurteilung zu gelangen, ob die funktionale Sicherheit durch die sicherheitsbezogenen E/E/PE-Systeme erreicht worden ist.

8.2.2 Diejenigen Personen, die die Beurteilung der funktionalen Sicherheit ausführen, müssen Zugriff auf alle Personen, die in irgendeine Tätigkeit des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus oder des Software-Sicherheitslebenszyklus eingebunden sind, und alle relevanten Informationen und Einrichtungen (sowohl Hardware als auch Software) haben.

8.2.3 Die Beurteilung der funktionalen Sicherheit muss in allen Phasen des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus und dem Software-Sicherheitslebenszyklus stattfinden. Diejenigen Personen, die die Beurteilung der funktionalen Sicherheit ausführen, müssen die während jeder Phase des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus und des Software-Sicherheitslebenszyklus ausgeführten Tätigkeiten und erzielten Ergebnisse betrachten und beurteilen, inwieweit die Ziele und Anforderungen dieser Norm erreicht worden sind.

8.2.4 Die Beurteilung der funktionalen Sicherheit muss während des gesamten Lebenszyklus, des E/E/PES-Lebenszyklus und des Software-Lebenszyklus ausgeführt werden. Sie darf unter Berücksichtigung der vorrangigen Anforderung, dass eine Beurteilung der funktionalen Sicherheit ausgeführt werden muss, bevor eine ermittelte Gefährdung auftritt, nach jeder Phase eines Sicherheitslebenszyklus oder nach mehreren Sicherheits-Lebenszyklusphasen ausgeführt werden.

Beispiel: ISO 26262 (abgeleitet aus DIN EN 61508)

- Sicherheitsnorm zur funktionalen Sicherheit von Kraftfahrzeugen
- Inkrafttreten November 2011 (aktuell Überarbeitung zu Version 2 in 2017/18)
- Im Beuth-Verlag erschienen, Kosten gesamt: 1.262,70 Euro
- Umfang ca. 400 Seiten in 10 Teilen auf engl. Sprache:
  - Teil 1: Vokabular
  - Teil 2: Management der funktionalen Sicherheit
    - Projektübergreifendes Sicherheitsmanagement
    - Sicherheitsmanagement während der Entwicklung
    - Aktivitäten nach Produktfreigabe

Beispiel: ISO 26262 (abgeleitet aus DIN EN 61508)

- Teil 3: Konzeptphase
  - Definition des Systems
  - Identifikation und Analyse von Gefährdungen
  - Risikoabschätzung
- Teil 4: Produktentwicklung auf Systemebene
- Teil 5: Produktentwicklung auf Hardwareebene
- Teil 6: Produktentwicklung auf Softwareebene

Beispiel: ISO 26262 (abgeleitet aus DIN EN 61508)

- Teil 7: Anforderungen an Produktion, Betrieb und Außerbetriebnahme
  - Installation und Stilllegung
  - Wartung, Pflege und Reparatur
- Teil 8: Unterstützende Prozesse
  - Konfigurations- und Änderungsmanagement
  - Verifikation und Dokumentation
- Teil 9: SIL- und sicherheitsorientierte Analysen
  - Unterteilung in Sicherheitsanforderungsstufen (SIL)
  - Möglichkeiten zur Reduzierung von Gefahrensituationen
- Teil 10: Guideline
  - Anwendungsbeispiele
  - Weiterführende Informationen (Erläuterungen)

Weitere Beispiele abgeleiteter Normen:

DIN EN 61511

Funktionale Sicherheit – Sicherheitstechnische Systeme für Prozessindustrie

Stand: 2005-05

DIN EN ISO 13849

Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen

Stand: 2008-12

- **Teil 1:** Allgemeine Gestaltungsgrundsätze
- **Teil 2:** Validierung

DIN EN ISO 50129

Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Sicherheitsrelevante elektronische Systeme für Signaltechnik

Stand: 2003-12

Weitere Beispiele abgeleiteter Normen:

VDI/VDE 3542

Sicherheitstechnische Begriffe für Automatisierungssysteme

Stand: 2000-10

Teil 1: Qualitative Begriffe

Teil 2: Quantitative Begriffe und Definitionen

Teil 3: Anwendungshinweise und Beispiele

Teil 4: Zuverlässigkeit und Sicherheit komplexer Systeme (Begriffe)

DIN 40041

Zuverlässigkeit; Begriffe

Stand: 1990-12

Frage zu Kapitel 1.2

Was ist der Kerninhalt der Norm ISO 26262?

Was ist der grundsätzliche Unterschied zwischen den Normen DIN EN 61508 und der Norm DIN EN 62061?

Gliederung

§ 2 Wahrscheinlichkeit und Zuverlässigkeit

2.1 Grundlagen der Wahrscheinlichkeitsrechnung

2.2 Lebensdauerverteilungen

2.3 Verfügbarkeit von Systemen

2.4 Zuverlässigkeitsblockdiagramm

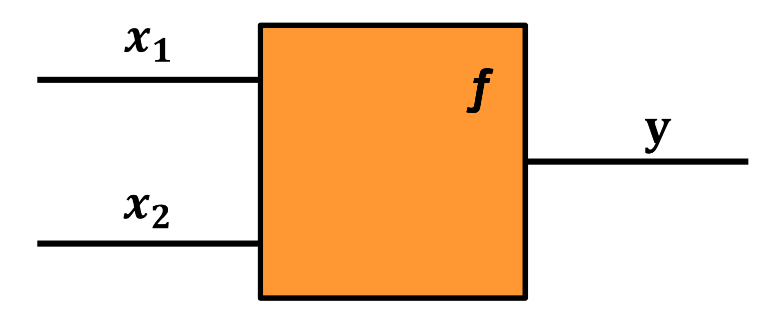
Einführung - Wahrscheinlichkeit zufälliger Ereignisse

- Wahrscheinlichkeitsrechnung beschreibt ein Zufallsexperiment, das unter bestimmten Randbedingungen durchgeführt wird und dessen Ergebnis über ein Ereignis E charakterisiert wird
- Beispiele für zufällige Ereignisse:
  - Werfen eines Würfels
  - Verhalten einer Komponenten innerhalb/außerhalb eines definierten Bereichs
  - Zustand eines Systems (funktionsfähig/nicht funktionsfähig)
- Wahrscheinlichkeit P(E) für die ein Ereignis E auftritt:

$$\mathbf{P(E)} = \frac{\textbf{Anzahl der zu E gehörenden Ergebnisse}}{\textbf{Anzahl aller Ergebnisse}}$$

Kombination von Ereignissen – Boolesche Modellbildung

- Ereignisse können miteinander verknüpft werden, um so zu neuen Ereignissen zu führen
- Verknüpfung erfolgt über logische Gatter, die als elementare Bausteine einer Schaltung logische Operationen durchführen
- Eine logische Operation ist eine boolesche Funktion über die Menge M = {0,1} und kann mithilfe von Wertetabellen definiert werden
- Ein Logikgatter repräsentiert dabei eine Funktion:



Grundgatter der Boolschen Modellbildung (1/3)

- Normung nach IEC 60617 – „Graphische Symbole für Schaltpläne“

<table>
<tr><th>Schaltzeichen (IEC 60617)</th><th>Wertetabelle</th><th>Funktion</th></tr>
<tr><td>Negation (NOT)
Eingang x_1, Symbol „1“, Ausgang y (negiert)</td><td>

x_1	y
1	0
0	1

</td><td>$\mathbf{y} = (\neg \mathbf{x_1})$
$\mathbf{y} = (\overline{\mathbf{x_1}})$</td></tr>
<tr><td>Disjunktion (OR)
Eingänge x_1, x_2, Symbol „≥ 1“, Ausgang y</td><td>

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

</td><td>$\mathbf{y} = (\mathbf{x_1} \vee \mathbf{x_2})$</td></tr>
</table>

Grundgatter der Boolschen Modellbildung (2/3)

Schaltzeichen (IEC 60617)	Wertetabelle	Funktion
Konjunktion (AND): Gatter „&“ mit Eingängen x_1, x_2 und Ausgang y	x_1 x_2 | y 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	$\mathbf{y} = (\mathbf{x_1} \wedge \mathbf{x_2})$ $\mathbf{y} = (\mathbf{x_1} * \mathbf{x_2})$
Antivalenz (XOR): Gatter „= 1“ mit Eingängen x_1, x_2 und Ausgang y	x_1 x_2 | y 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	$\mathbf{y} = (\mathbf{x_1} \oplus \mathbf{x_2})$

Grundgatter der Boolschen Modellbildung (3/3)

Schaltzeichen (IEC 60617)	Wertetabelle	Funktion
NAND: Eingänge x_1, x_2; Symbol „&“; negierter Ausgang y	x_1 x_2 | y: 0 0 | 1; 0 1 | 1; 1 0 | 1; 1 1 | 0	$y = (\overline{x_1 \wedge x_2})$ $y = (\overline{x_1 * x_2})$
NOR: Eingänge x_1, x_2; Symbol „≥ 1“; negierter Ausgang y	x_1 x_2 | y: 0 0 | 1; 0 1 | 0; 1 0 | 0; 1 1 | 0	$y = (\overline{x_1 \vee x_2})$

NAND:

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

NOR:

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

Boolesche Modellbildung – Axiome der Booleschen Algebra

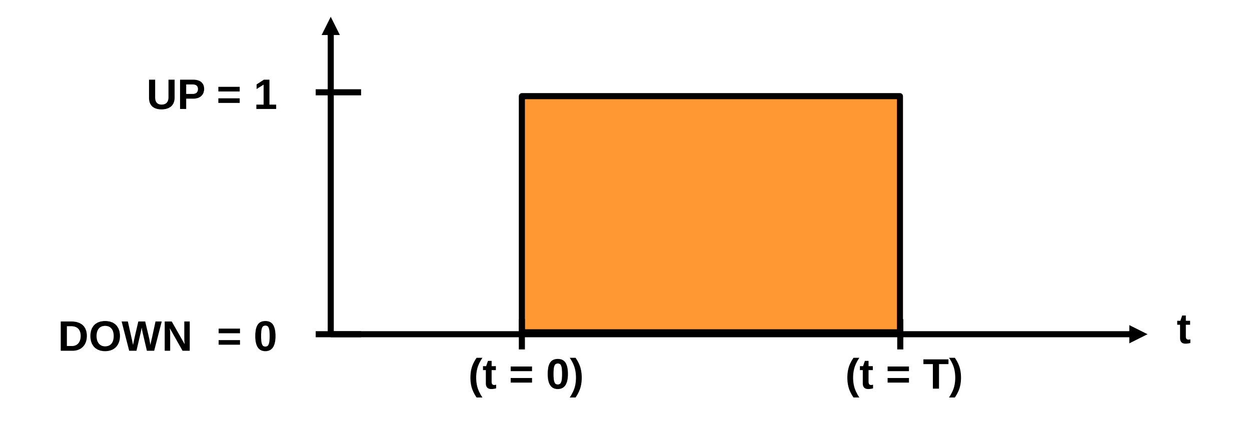
- Kommutativgesetz:
  $$(\mathbf{x_1} \land \mathbf{x_2}) = (\mathbf{x_2} \land \mathbf{x_1})$$
  $$(\mathbf{x_1} \lor \mathbf{x_2}) = (\mathbf{x_2} \lor \mathbf{x_1})$$
- Assoziativgesetz:
  $$\mathbf{x_1} \land (\mathbf{x_2} \land \mathbf{x_3}) = (\mathbf{x_1} \land \mathbf{x_2}) \land \mathbf{x_3}$$
  $$\mathbf{x_1} \lor (\mathbf{x_2} \lor \mathbf{x_3}) = (\mathbf{x_1} \lor \mathbf{x_2}) \lor \mathbf{x_3}$$
- Distributivgesetz:
  $$\mathbf{x_1} \lor (\mathbf{x_2} \land \mathbf{x_3}) = (\mathbf{x_1} \lor \mathbf{x_2}) \land (\mathbf{x_1} \lor \mathbf{x_3})$$
  $$\mathbf{x_1} \land (\mathbf{x_2} \lor \mathbf{x_3}) = (\mathbf{x_1} \land \mathbf{x_2}) \lor (\mathbf{x_1} \land \mathbf{x_3})$$
- Idempotenzgesetz:
  $$(\mathbf{x} \land \mathbf{x}) = \mathbf{x}$$
  $$(\mathbf{x} \lor \mathbf{x}) = \mathbf{x}$$
- De Morgansches Gesetz:
  $$(\overline{\mathbf{x_1} \lor \mathbf{x_2}}) = (\overline{\mathbf{x_1}} \land \overline{\mathbf{x_2}})$$
  $$(\overline{\mathbf{x_1} \land \mathbf{x_2}}) = (\overline{\mathbf{x_1}} \lor \overline{\mathbf{x_2}})$$
- Weiter gilt:
  $$(\mathbf{x} \lor \mathbf{0}) = \mathbf{x} \qquad (\mathbf{x} \lor \mathbf{1}) = \mathbf{1} \qquad (\mathbf{x} \lor \bar{\mathbf{x}}) = \mathbf{1}$$
  $$(\mathbf{x} \land \mathbf{1}) = \mathbf{x} \qquad (\mathbf{x} \land \mathbf{0}) = \mathbf{0} \qquad (\mathbf{x} \land \bar{\mathbf{x}}) = \mathbf{0}$$

Zuverlässigkeit als Wahrscheinlichkeit

- Quantitative Methoden sind eng mit Begriffen und Verfahren der Statistik und Wahrscheinlichkeit verknüpft
- Ausfallzeiten von Bauteilen sind gewisser Stochastik und Streuung unterworfen
- Einflussnahme auf die Ursachen der Streuung sind begrenzt
- Ein Ausfall lässt sich daher nicht über rein deterministische Verfahren bestimmen
- Kenntnis der Wahrscheinlichkeitsrechnung notwendig, u.a.:
  - Ausfallrate
  - Ausfalldichte
  - Ausfallwahrscheinlichkeit
  - Überlebenswahrscheinlichkeit / Zuverlässigkeit

Zustand eines technischen Systems

– Zustand des Systems zum Zeitpunkt t:



– Lebensdauer T eines technischen Systems ist die Zeitspanne für t ≥ 0, von der (ersten) Inbetriebnahme bei t = 0 bis hin zum Ausfall des Systems bei t = T

$$\mathbf{X}(\mathbf{t}) = \begin{cases} \mathbf{1}, & \mathbf{t} < \mathbf{T} \\ \mathbf{0}, & \mathbf{t} \geq \mathbf{T} \end{cases}$$

Ausfallwahrscheinlichkeit F(t)

- Lebensdauer T ist eine reelle Zufallsgröße mit einer gewissen Verteilungsfunktion
- Verteilungsfunktion heißt Ausfallwahrscheinlichkeit F(t) und stellt die Wahrscheinlichkeit P so dar, dass die Zufallsgröße T kleiner oder gleich eines vorgegebenen Wertes t ist (*probability of failure*) :

$$\mathbf{F(t) = P(T \leq t)}$$

t

- F(t) ist die Wahrscheinlichkeit für das Eintreten eines Systemausfalls im Intervall [0,t] mit:

Ausfalldichte f(t)

– Häufigkeit der Ausfälle, d.h., die zeitliche Ableitung der Ausfallwahrscheinlichkeit wird als Ausfalldichte f(t) bezeichnet (*failure density*) :

$$\mathbf{f(t) = \frac{dF(t)}{dt}}$$

– Für die Ausfalldichte gilt:

$$\mathbf{f(t)\ ist} \begin{cases} = \mathbf{0}, & \mathbf{t < 0} \\ \geq \mathbf{0}, & \mathbf{t \geq 0} \end{cases}$$

Überlebenswahrscheinlichkeit / Zuverlässigkeit R(t)

– Zuverlässigkeit R(t) stellt die Wahrscheinlichkeit P so dar, dass die Zufallsgröße T größer eines vorgegebenen Wertes t ist (*reliability*):

$$\mathbf{R(t) = P(T > t)}$$

t

– R(t) ist die Wahrscheinlichkeit für keinen Systemausfall im Intervall [0,t], d.h., die Zuverlässigkeit ist das Gegenereignis zur Ausfallwahrscheinlichkeit mit:

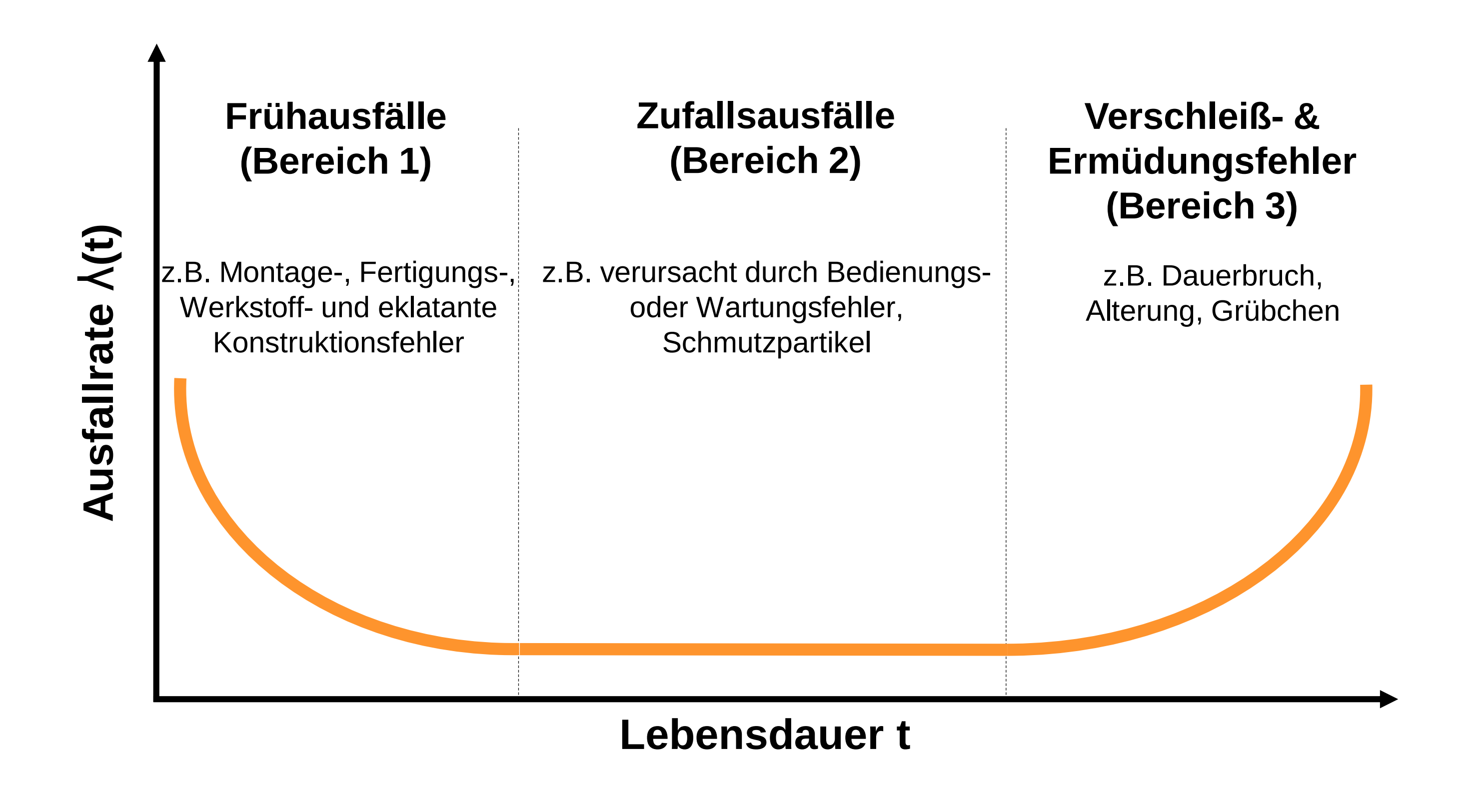
$$\mathbf{R(t = 0) = 1}$$

$$\mathbf{R(t \rightarrow \infty) = 0}$$

Ausfallrate λ(t)

– Ausfallrate λ(t) ist die Ausfallneigung eines Systems in Abhängigkeit der Zeit

– Kenngröße, die die Wahrscheinlichkeit eines Systemausfalls bezogen auf die gesamte Lebensdauer T darstellt (*failure rate*):



Ausfallraten von Komponenten (1/2)

– Unternehmen und Forschungsinstitute veröffentlichen Datenbanken oder (Online-) Handbücher zur Bestimmung der Ausfallraten von Komponenten:

- Siemens (SN29500)
- Union Technique de l'Electricite (RDF 2000)
- Reliability Information Center (EPRD-97)

– Beispiel: EPRD-97

Part Description	Quality Level	App. Env.	Data Source	Fail. Rate Fails/(E6)	Total Failed	Op. Hours/ Miles (E6)
Capacitor, Fixed, Electrolytic (Summary)				0.1745		
	Commercial	GBC		0.0070		
	Military			0.3410		
		AIA		< 0.1669		
		AIC		0.1847		
		AU		0.2149		
		AUA		2.3770		
		AUF		5.3966		
		G		0.7143		
		GF		0.0957		
Capacitor, Fixed, Electrolytic				1.1228		
	Military			1.1228		
		AIA	23035-000	< 0.1728	0	5.7865
		AIC	17189-000	< 0.3166	0	3.1584
		AU	13655-000	0.2200	85	386.3482
		AUA	23035-000	2.4194	28	11.5731
		AUF	23035-000	5.4930	36	6.5538
		GF	14851-000	0.5899	17	28.8183
Capacitor, Fixed, Electrolytic, Al				0.0455		
	Commercial	GBC	13567-021	0.0099	236	23852.2128
	Military			0.0859		
		AU	13655-000	< 0.1091	0	9.1624
		AUA	23035-000	< 4.8388	0	0.2067
		AUF	23035-000	< 8.5447	0	0.1170
		GF		0.0976		
			14851-000	0.2082	10	48.0305
			23039-000	0.0458	1	21.8542
Capacitor, Fixed, Electrolytic, Ta				0.0094		
	Commercial	GBC	13567-021	0.0049	224	45341.4884
	Military	GF	14851-000	0.0189	3	166.5056
Capacitor, Fixed, Electrolytic, Ta Foil				0.7143		
	Military	G	23040-000	0.7143	5	7.0000
Capacitor, Fixed, Electrolytic, Ta Solid				0.0655		
	Military			0.0655		
		AIA	23035-000	< 4.8388	0	2.2067
		AIC	17189-000	0.4433	1	2.2560
		GF	14851-000	< 0.0781	0	12.8081

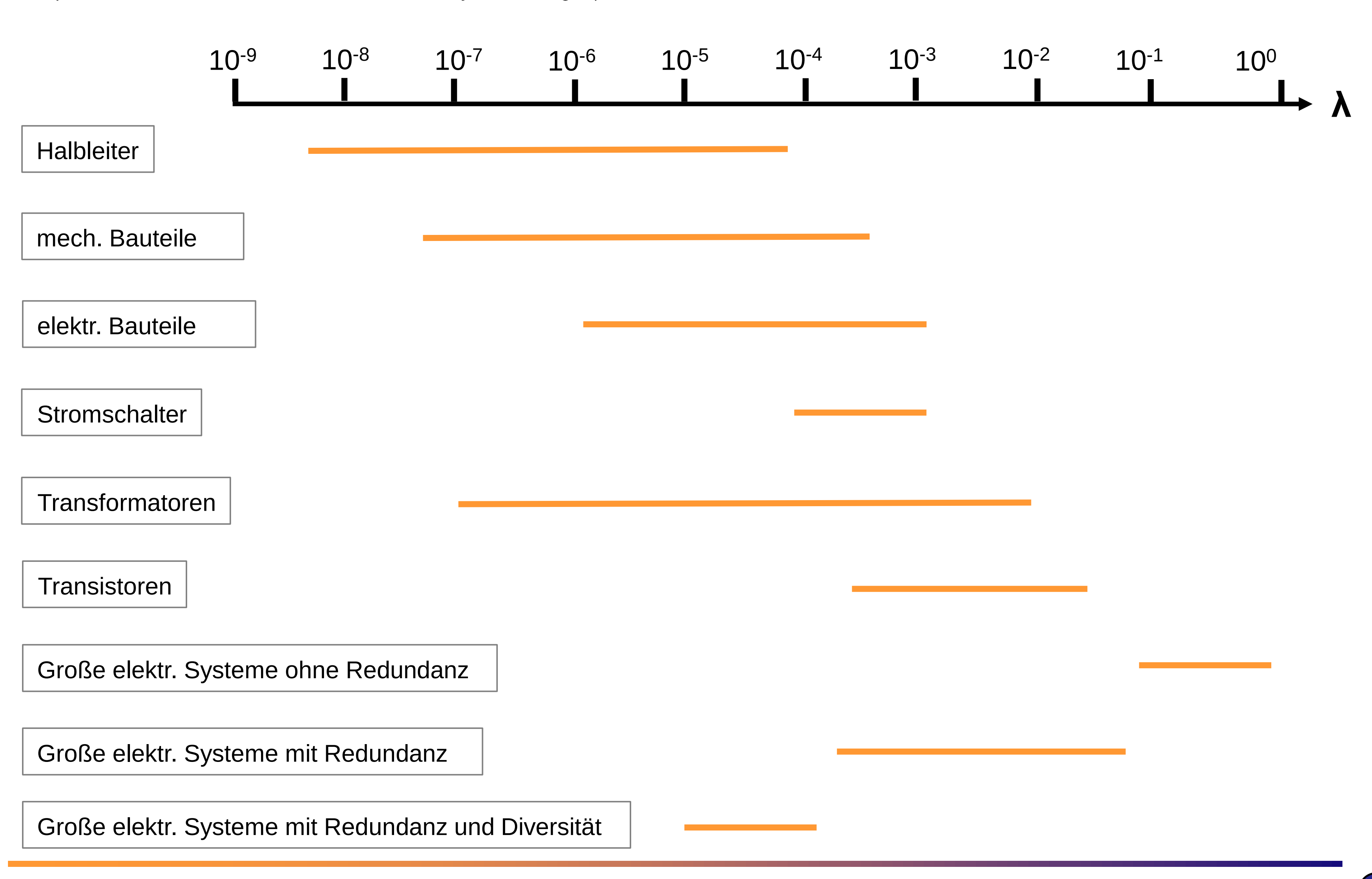
Quelle: www.theriac.org

Ausfallraten von Komponenten (2/2)

- Bei den in Dokumenten genannten Werte handelt es sich um bekannte, konstante Ausfallraten für Komponenten, die mit Qualitäts- und Einsatzfaktoren modifiziert werden um Unsicherheiten zu berücksichtigen
- Die Angaben sind daher in der Regel (erheblich) größer als im konkreten Einsatz und variieren für bestimmte Komponenten je nach Herausgeber deutlich
- Relativ gesicherte Angaben für eine bestimmte Komponente können direkt den „*Reliability Reports*“ der Hersteller entnommen werden
- In 1970er und 1980er versuchte die Projekt-Gruppe EuReDatA (*European Reliability Data Banks Association*), den Austausch und die Zusammenfassung von Datenbeständen zwischen Unternehmen, Behörden und Hochschulen zu fördern

Typische Ausfallratenbereiche

(nach A.E. Green und A.J. Bourne : „Reliability Technologie“)



Frage zu Kapitel 2.1

Welchen Aussagen stimmen Sie zu?

☐ Die Zuverlässigkeit eines Bauteils weist direkt bei Inbetriebnahme eine Wahrscheinlichkeit von 100% auf.

☐ Mit steigender Ausfallwahrscheinlichkeit sinkt die Zuverlässigkeit.

☐ In frühen Stadien eines Produkts ist mit einer hohen Ausfallrate zu rechnen.

☐ Ausfälle lassen sich über rein deterministische Verfahren berechnen.

§ 2 Wahrscheinlichkeit und Zuverlässigkeit

Einführung

- Lebensdauer ist die Zeit, in der eine Einheit betriebsbereit ist
- Lebensdauer selbst besitzt eine Lebensdauerverteilung, die als quantitative Methode zur Prognose der Zuverlässigkeit verwendet wird
- Kontinuierliche Lebensdauerverteilungen:
  - Exponentialverteilung
  - Weibullverteilung
  - Erlang-Verteilung
- Diskrete Lebensdauerverteilungen:
  - Binomialverteilung
  - Poisson-Verteilung

Exponentialverteilung – Allgemeines


Quelle: www.heise.de, 2014

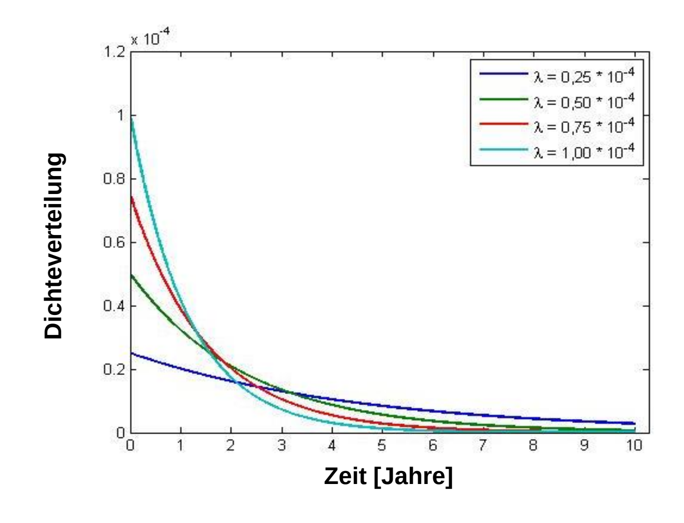
- Moderne Wahrscheinlichkeitsrechnung wurde 1933 von dem russischen Mathematiker Andrei Kolmogorow (1903 - 1987) in seinem Buch „Grundbegriffe der Wahrscheinlichkeitsrechnung“ beschrieben
- 1920-1925 Studium am chemisch technischen Institut der staatlichen Lomonossow-Universität Moskau
- 1929 Promotion, 1931 Professur auf dem Gebiet der Wahrscheinlichkeit
- 1933 Direktor des mathematischen Instituts
- Exponentialverteilung beschreibt eine stetige Wahrscheinlichkeitsverteilung über die Menge der positiven reellen Zahlen.
- Lebensdauer von elektrischen Bauteilen und Systemen mit sehr geringer Alterung bzw. Verschleiß

Exponentialverteilung – Gleichungen

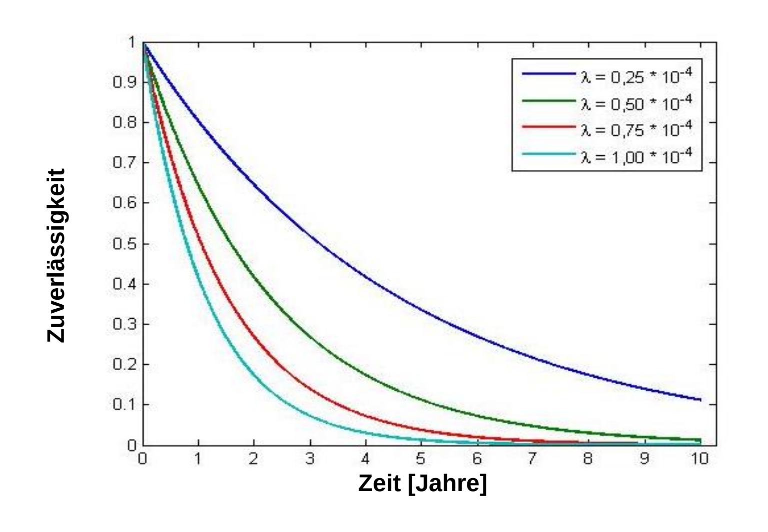
- Annahme einer konstanten Ausfallrate, d.h., die Restlebensdauer hängt nicht von der bisherigen Lebensdauer ab
- Es gilt für $t \geq 0$:
  - Ausfallrate: $\boldsymbol{\lambda(t) = \lambda_0 = \mathrm{const.}}$
  - Ausfalldichte: $\boldsymbol{f(t) = \lambda_0 * e^{-\lambda_0 t}}$
  - Ausfallwahrscheinlichkeit: $\boldsymbol{F(t) = 1 - e^{-\lambda_0 t}}$
  - Zuverlässigkeit: $\boldsymbol{R(t) = e^{-\lambda_0 t}}$

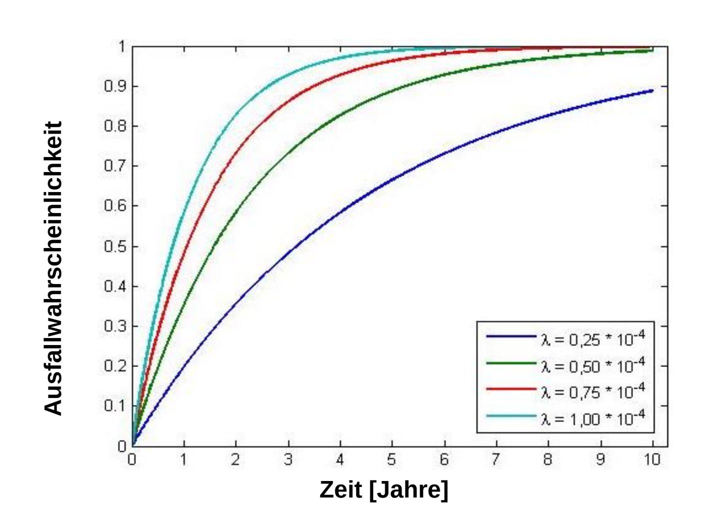
Exponentialverteilung – Simulation

– Ausfalldichte:



– Zuverlässigkeit und Ausfallwahrscheinlichkeit:





Exponentialverteilung - Rechenbeispiel

Die Lebensdauer eines Bauteils sei exponentiell-verteilt mit $\lambda_0 = 0{,}5 \cdot 10^{-5}\,h^{-1}$.

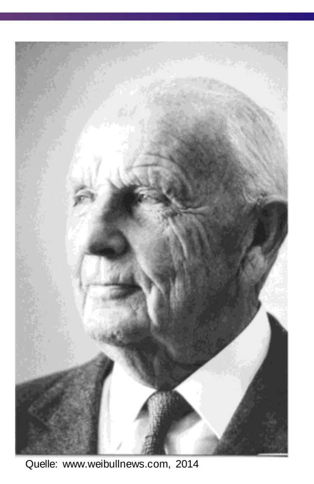
a) Wie groß ist die Wahrscheinlichkeit, dass das Bauteil innerhalb eines Jahres ausfällt?

$$\mathbf{P(T < 1a) = F(1a) = F(8.760h) = 1 - e^{-0{,}5 * 10^{-5} h^{-1} * 8.760h} = 0{,}0428 \approx 4{,}3\%}$$

b) Wie große ist die Wahrscheinlichkeit, dass das Bauteil zwischen dem 5. und 10. Jahr ausfällt?

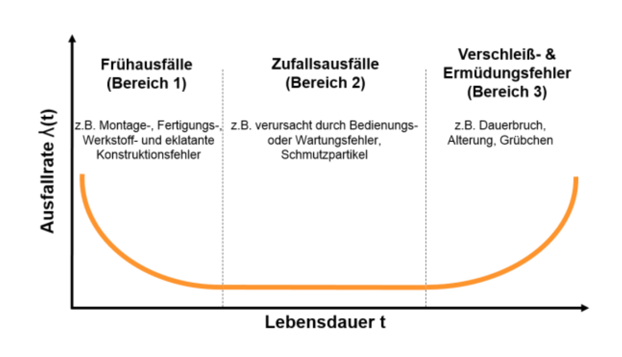
$$\mathbf{P(5a \leq T \leq 10a) = F(10a) - F(5a) = 0{,}355 - 0{,}197 = 0{,}158 \approx 15{,}8\%}$$

Weibullverteilung - Allgemeines


Quelle: www.weibullnews.com, 2014

- Benannt nach Waloddi Weibull (1887 – 1979)
- Schwedischer Mathematiker und Ingenieur
- 1932 Promotion an der Universität Uppsala
- 1941 Professur für technische Physik an der königlichen technischen Hochschule Stockholm
- Verfasste diverse Abhandlungen über die Materialfestigkeit, Materialermüdung und das Bruchverhalten von Festkörpern
- Veröffentlichung der Weibull-Verteilung in 1951:
    - Beschreibung von Lebensdauerverteilungen mit monoton fallender, konstanter und monoton wachsender Ausfallrate möglich
    - Findet Einsatz bei lebensdauerbeeinflussenden Lastmerkmalen wie Spannung oder Kraft, sowie bei dynamischen Festigkeitsversuchen

Weibullverteilung – Beschreibung



– Charakterisierung über Parameter:
  - Ausfallsteilheit bzw. Formparameter b
    - Frühausfälle (1) mit $b < 1$
    - Zufallsausfälle (2) mit $b = 1$
    - Verschließ-/Ermüdungs-Fehler (3) mit $b > 1$



  - Charakteristische Lebensdauer bzw. Lageparameter T
    - Praxis typ. Zeitspanne, in der 63% aller Komponenten (Einheiten eines Systems, Bauteile einer Serienproduktion,…) ausfallen

– Beispiel für Komponenten mit typischem
  - Frühausfall-Verhalten:
  - Verschleiß-/Ermüdungs-Fehler-Verhalten:

Weibullverteilung – Gleichungen

– Es gilt für $t \geq 0$:

- Ausfallrate: $$\boldsymbol{\lambda(t) = \frac{b}{T} * \left(\frac{t}{T}\right)^{b-1}}$$

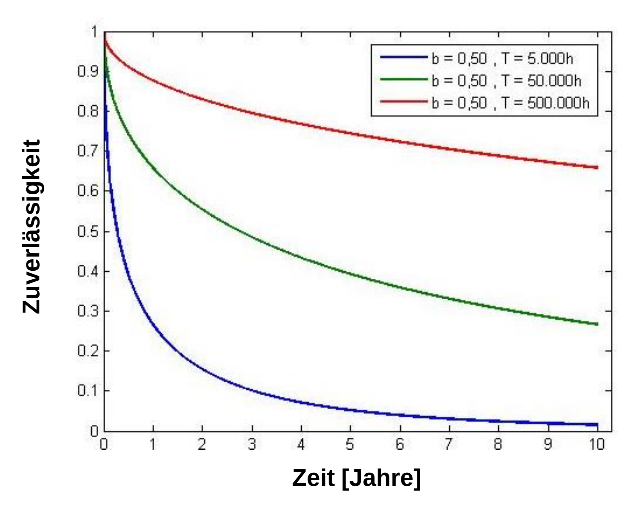
- Ausfalldichte: $$\mathbf{f(t) = \frac{b}{T} * \left(\frac{t}{T}\right)^{b-1} * e^{\left[-\left(\frac{t}{T}\right)^{b}\right]}}$$

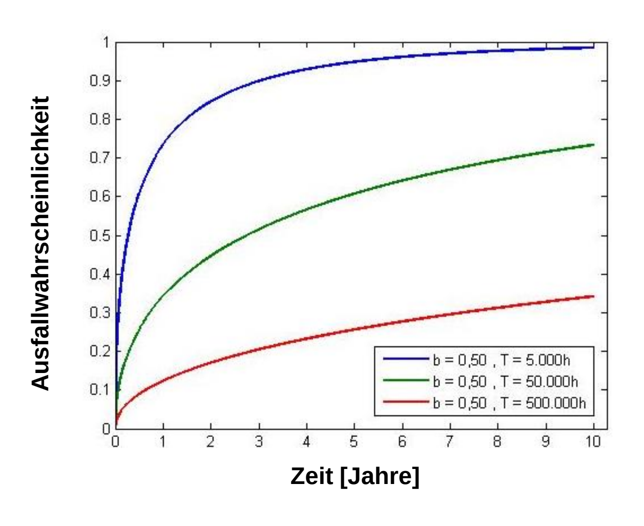
- Ausfallwahrscheinlichkeit: $$\mathbf{F(t) = 1 - e^{\left[-\left(\frac{t}{T}\right)^{b}\right]}}$$

- Zuverlässigkeit: $$\mathbf{R(t) = e^{\left[-\left(\frac{t}{T}\right)^{b}\right]}}$$

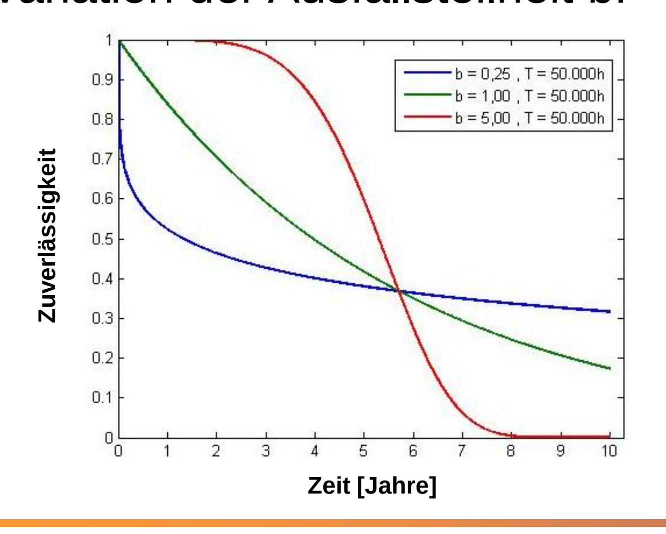
Weibullverteilung - Simulation

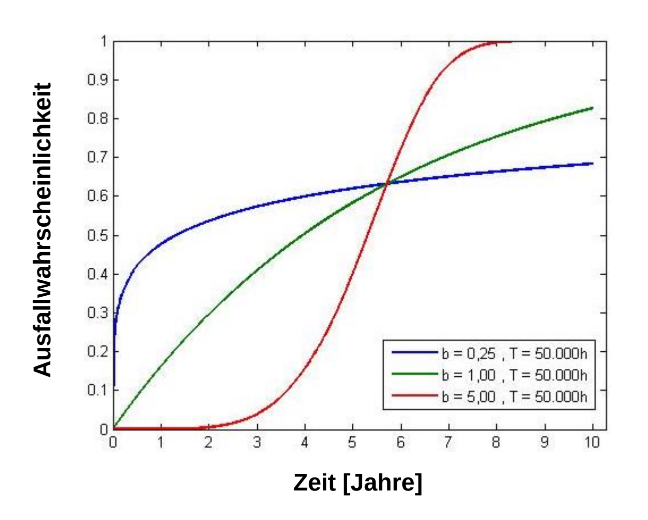
– Variation der charakteristischen Lebensdauer T:





– Variation der Ausfallsteilheit b:





Weibullverteilung - Rechenbeispiel

Ein System sei weibull-verteilt mit dem Formparameter b = 2.

Welche charakteristische Lebensdauer T liegt vor, wenn das System im ersten Jahr eine Zuverlässigkeit von 95% aufweist?

$$\mathbf{R(8.760h) = e^{\left[-\left(\frac{8.760h}{T}\right)^{2}\right]} = 0,95}$$

$$\mathbf{\left(\frac{8.760h}{T}\right)^{2} = -\ln(0,95)}$$

$$\mathbf{T = \frac{8.760h}{\sqrt{-\ln(0,95)}} \approx 38.679h}$$

Erlang-Verteilung - Allgemeines


Quelle: de.wikipedia.org, 2014

- Entwickelt von Agner Krarup Erlang (1878 – 1929)
- Dänischer Mathematiker und Ingenieur
- 1896 – 1901 Studium an der Universität Kopenhagen
- Nach verschiedenen Lehrtätigkeiten ab 1908 Entwicklungsingenieur bei der Kopenhagener Telefongesellschaft
- Veröffentlichung der Erlang-Verteilung in 1917 zur Beschreibung der statistischen Modellierung von Intervall-Längen zwischen Telefonanrufen
- Einsatz der Erlang-Verteilung:
  - Vorwiegend im Zusammenhang mit der Warteschlangen- bzw. Bedienungs-Theorie und Schalt-Redundanzen
  - Hier: Betrachtung von n-Stufen mit derselben konstanten Ausfallrate λ_0

Erlang-Verteilung - Gleichungen

– Es gilt für $t \geq 0$:

- Ausfallrate: $\boldsymbol{\lambda(t) \stackrel{\text{def}}{=} \lambda_0}$

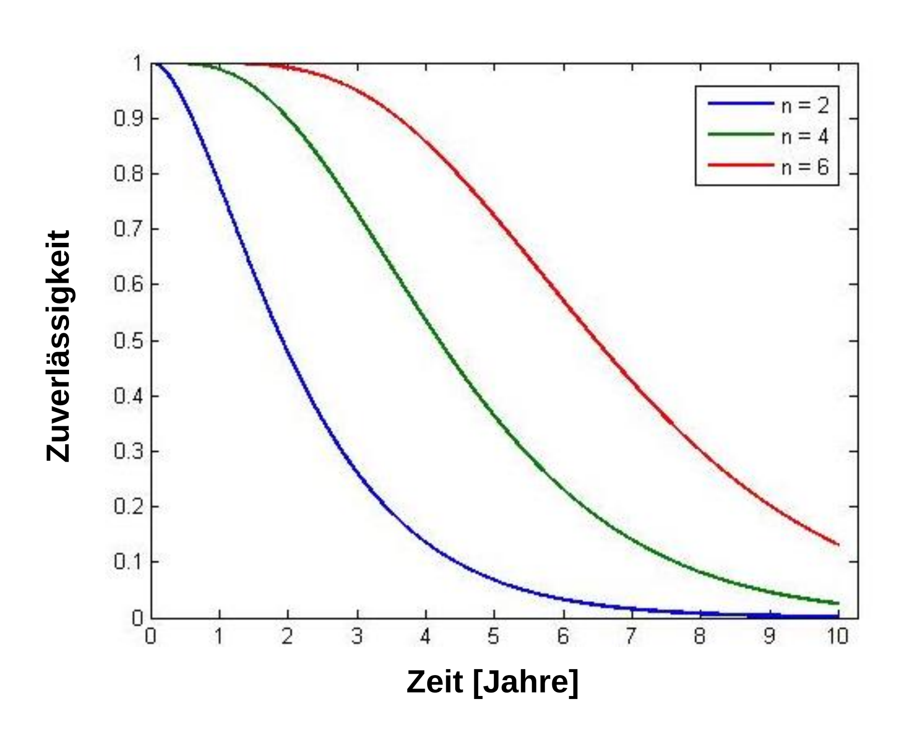
- Ausfalldichte: $\boldsymbol{f(t) = \lambda_0 * \frac{(\lambda_0 * t)^{n-1}}{(n-1)!} * e^{-\lambda_0 t}}$

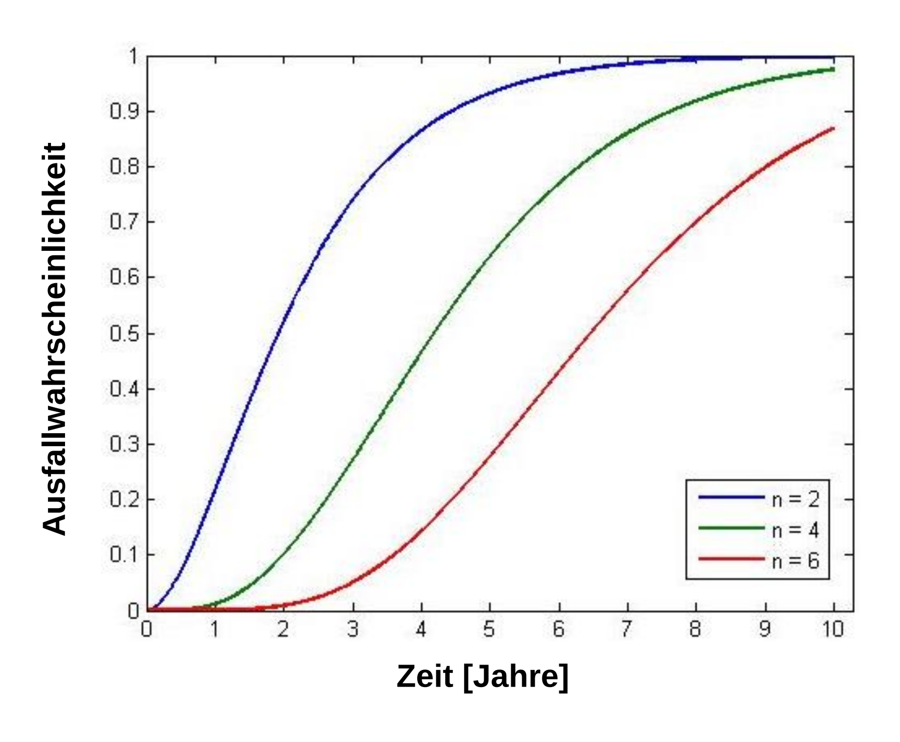
- Ausfallwahrscheinlichkeit: $\boldsymbol{F(t) = 1 - e^{-\lambda_0 t} * \sum_{i=0}^{n-1} \frac{(\lambda_0 * t)^i}{i!}}$

- Zuverlässigkeit: $\boldsymbol{R(t) = e^{-\lambda_0 t} * \sum_{i=0}^{n-1} \frac{(\lambda_0 * t)^i}{i!}}$

Erlang-Verteilung – Simulation

– Betrachtung von verschiedenen n-Stufen von Schaltredundanzen





Erlang-Verteilung - Rechenbeispiel

Zur Sicherstellung der Stromversorgung einer Anlage werden zwei gleiche Generatoren als Standby-System mit einer Ausfallrate von $\lambda_0 = 10^{-4}\ h^{-1}$ eingesetzt. Die Generatoren werden alle 14 Tage gewartet. Welche Zuverlässigkeit liegt innerhalb der Wartungsintervalle vor?

$$\mathbf{R(t) = e^{-\lambda_0 t} * \sum_{i=0}^{n-1} \frac{(\lambda_0 * t)^i}{i!}}$$

$$\mathbf{R(t) = e^{-\lambda_0 t} * \sum_{i=0}^{2-1} \frac{(\lambda_0 * t)^i}{i!} = e^{-\lambda_0 t} * \left(1 + \frac{\lambda_0 * t}{1!}\right)}$$

$$\mathbf{= e^{-10^{-4} h^{-1} 336 h} * \left(1 + \frac{10^{-4} h^{-1} * 336 h}{1!}\right) = 0,9994 = 99,94\%}$$

Binomialverteilung - Allgemeines



Quelle: de.wikipedia.org, 2014

- Erste Diskussion der Verteilung von Blaise Pascal (1623 – 1662)
- Französischer Mathematiker, Physiker und Philosoph
- 1654 „Geburtsstunde der Wahrscheinlichkeitsrechnung"



  - Untersuchung der Gewinnchancen bei Würfelspielen

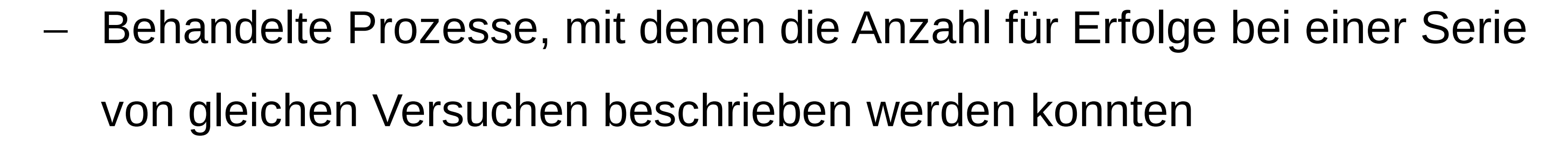


  - Veröffentlichung einer Abhandlung mit Beschreibung der Binomialkoeffizienten



Quelle: de.wikipedia.org, 2014

- Vollständige Beschreibung von Jakob I. Bernoulli (1655 – 1705)
- Schweizer Mathematiker und Physiker
- Baute seine Abhandlungen auf den Diskussionen Pascals auf



- Behandelte Prozesse, mit denen die Anzahl für Erfolge bei einer Serie von gleichen Versuchen beschrieben werden konnten

Binomialverteilung - Beschreibung

– Diskrete Lebensdauerverteilung, die bei einem Experiment mit der Erfolgswahrscheinlichkeit bzw. dem Merkmal p und der Anzahl bzw. Serie von gleichartigen und unabhängigen Versuchen n beschreibt, mit welcher Wahrscheinlichkeit genau k Erfolge erzielt werden können

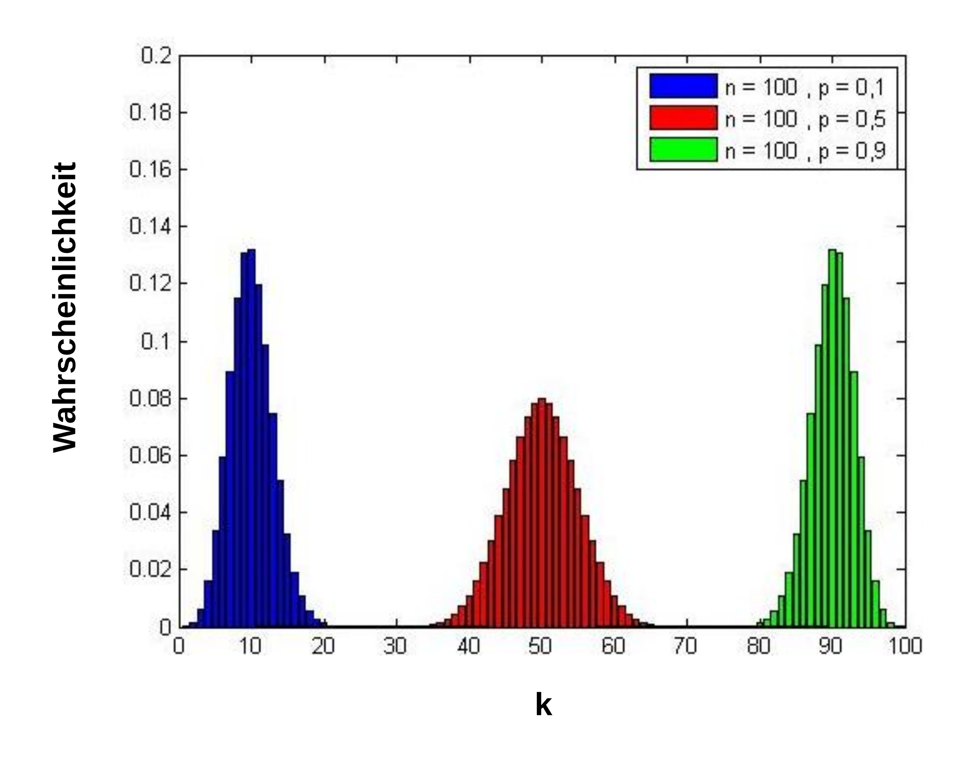
– Es gilt für eine Zufallsvariable X und n ≥ k mit n, k ≥ 0:

- Verteilungsdichte: $\boldsymbol{P(X = \mathbf{k}) = \binom{\mathbf{n}}{\mathbf{k}} * \mathbf{p}^{\mathbf{k}} * (\mathbf{1} - \mathbf{p})^{\mathbf{n}-\mathbf{k}}}$

- Binomialkoeffizient: $\boldsymbol{\binom{\mathbf{n}}{\mathbf{k}} = \frac{\mathbf{n}!}{\mathbf{k}! * (\mathbf{n} - \mathbf{k})!}}$

Binomialverteilung – Simulation

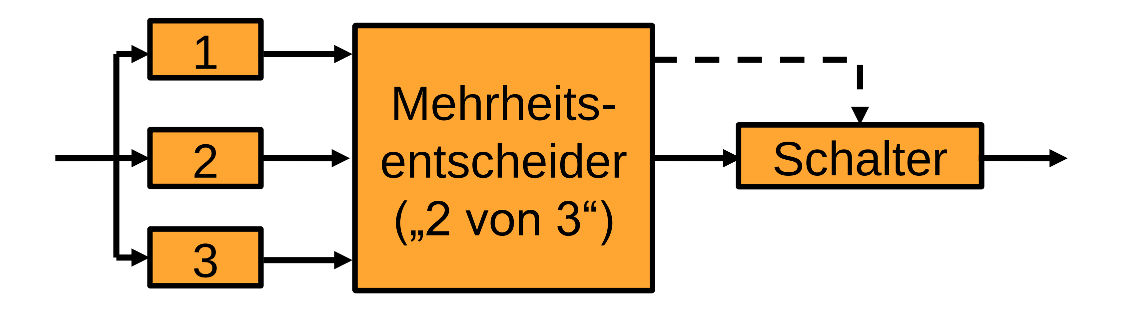
- Verteilungsdichte ordnet den natürlichen Zahlen k bestimmte Wahrscheinlichkeitswerte zu
- Simulation der Wahrscheinlichkeitsverteilung bei n = 100 Versuchen und veränderlicher Erfolgswahrscheinlichkeit p



Binomialverteilung – Zuverlässigkeitsberechnung

– Einsatz zur Berechnung der Zuverlässigkeit von „mvn–Systemen“

– Redundante Systeme (Majoritätsredundanz), bei denen m von n Komponenten funktionieren müssen (z.B. 2 von 3)



– Voraussetzung ist, dass dabei alle Komponenten dieselbe Ausfallrate besitzen und somit identische Zuverlässigkeitswerte p aufweisen

– Es gilt:

$$\mathbf{R_{mvn}} = \sum_{\mathbf{k=m}}^{\mathbf{n}} \left[\binom{\mathbf{n}}{\mathbf{k}} * \mathbf{p^k} * (\mathbf{1} - \mathbf{p})^{\mathbf{n-k}} \right]$$

Poisson-Verteilung - Allgemeines



Quelle: de.wikipedia.org, 2014

- Benannt nach Siméon Denis Poisson (1781 – 1840)
- Französischer Mathematiker und Physiker
- 1798 – 1800 Studium der Mathematik an der Ecole Polytechnique in Paris
- 1802 stellvertretender Professor von Joseph Fourier
- 1806 Übernahme dessen Lehrstuhl für Analysis und Mechanik
- 1837 Veröffentlichung erster Gedankengänge zur Poisson-Verteilung (im Rahmen der Wahrscheinlichkeitsbetrachtung von Urteilen in Straf- und Zivil-Sachen):
  - Basiert auf der Binomialverteilung und stellt deren Grenzverteilung dar
  - Verteilung für einen großen Stichprobenumfang bzw. Zahl der Versuche $n \rightarrow \infty$ und kleine Merkmalswerte bzw. Abnahme der Wahrscheinlichkeit $p \rightarrow 0$

Poisson-Verteilung - Beschreibung

– Die Forderung liegt vor, dass das Produkt aus n und p weder Null noch Unendlich wird und gegen einen endlichen Grenzwert μ konvergiert, gemäß:

– Es gilt für eine Zufallsgröße X im Raum der natürlichen Zahlen k und dem Parameter $\mu > 0$:

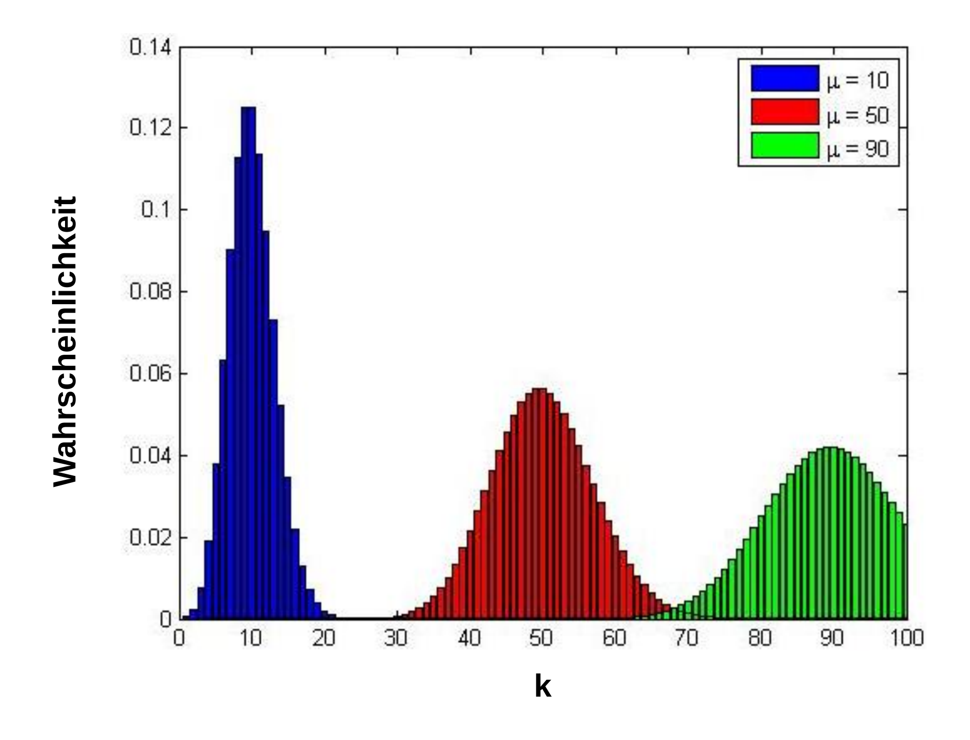
- Verteilungsdichte: $$\mathbf{P(X = k) = P_{\mu}(k) = \frac{\mu^k}{k!} * e^{-\mu}}$$

– Einsatz im Rahmen statistischer Prüfplanung von Ausfällen in der Produktion oder der Modellierung von Störfällen komplexer Industrieanlagen

– Experimente, bei denen ein Ereignis selten eintritt, dieses jedoch ein großes Risiko mit sich bringen kann

Poisson-Verteilung – Simulation

- Poisson-Verteilung wird vollständig über den Parameter μ beschrieben
- Simulation der Wahrscheinlichkeitsverteilung bei veränderlichem Parameter μ



Poisson-Verteilung - Rechenbeispiel

Ein Hersteller produziert Bauteile mit einem durchschnittlichen Ausschuss von 1%. Wie groß ist die Wahrscheinlichkeit, dass bei einer Qualitätskontrolle von 10 Bauteilen genau 2 defekt sind?

$$\mathbf{\mu = n * p = 10 * 0,01 = 0,1}$$

$$\mathbf{P(X = 2) = P_{0,1}(2) = \frac{0,1^2}{2!} * e^{-0,1} \approx 0,0045 = 0,45\%}$$

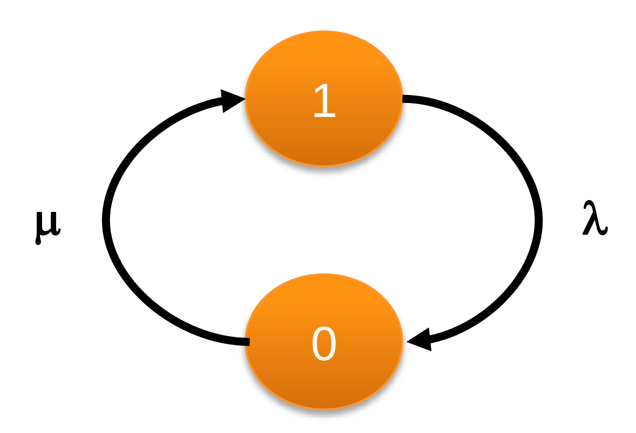
Frage zu Kapitel 2.2

Ein Sensors soll bei einer Betriebszeit von 10.000h eine exponentiell-verteilte Ausfallwahrscheinlichkeit von 8% nicht überschreiten. Welche dabei als konstant annehmbare Ausfallrate sollte der Sensor maximal haben?

§ 2 Wahrscheinlichkeit und Zuverlässigkeit

Verfügbarkeit

– Wahrscheinlichkeit, ein reparierbares System zu einem vorgegebenen Zeitpunkt in einem funktionsfähigem Zustand anzutreffen
– Zeitlicher Anteil der Benutzbarkeit eines Systems
  - Wert 1: System ist funktionsfähig
  - Wert 0: System ist nicht funktionsfähig



Ableitung des Erwartungswerts

– Der Erwartungswert ist der geometrische Mittelwert einer Zufallsgröße

– In der Zuverlässigkeitstechnik ist der Erwartungswert der Zeitpunkt, bei dem durchschnittlich ein Ausfall auftritt:

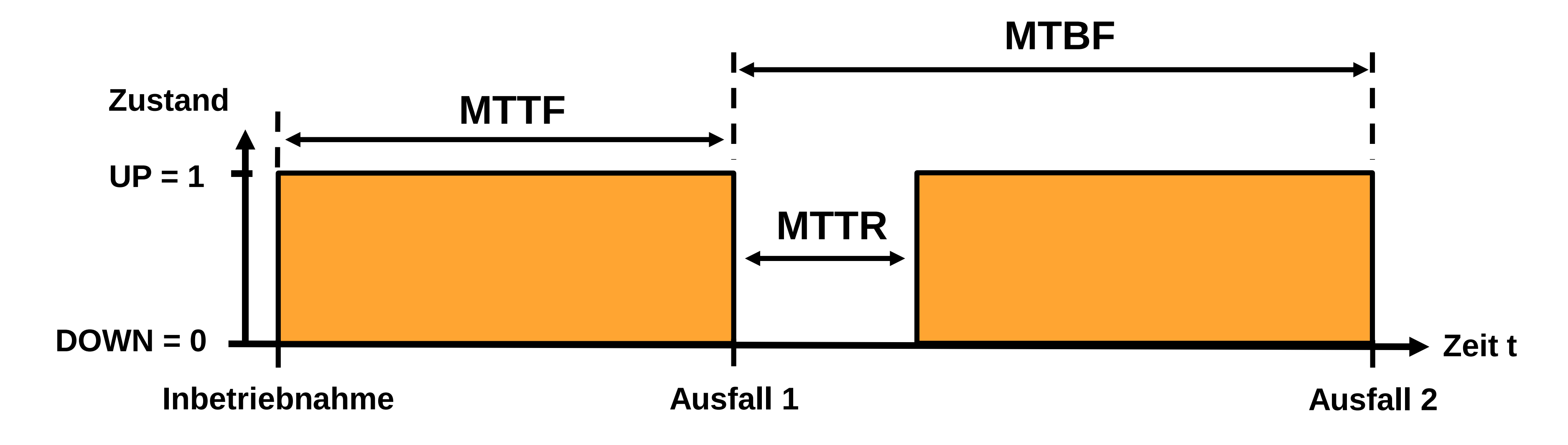
„The expectation of the time to failure“ (IEC 60050)

– Der Erwartungswert der Zuverlässigkeit wird als **MTTF** (**M**ean **T**ime **T**o **F**ailure) bezeichnet:

$$\mathbf{MTTF} = \mathbf{E}(\mathbf{t}) = \int_{\mathbf{0}}^{\infty} \mathbf{t} * \mathbf{f}(\mathbf{t})\mathbf{dt} = \int_{\mathbf{0}}^{\infty} \mathbf{R}(\mathbf{t})\mathbf{dt}$$

– Unter der Annahme einer konstanten Ausfallrate gilt:

Kenngrößen der Verfügbarkeit



- **MTTF** (**M**ean **T**ime **T**o **F**ailure):
  („Durchschnittliche Zeit bis zum Ausfall")

$$\mathbf{MTTF} = \frac{1}{\lambda}$$

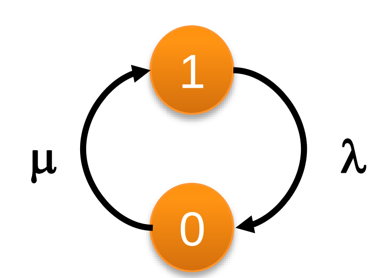
- **MTTR** (**M**ean **T**ime **T**o **R**epair):
  („Durchschnittliche Reparaturzeit")

$$\mathbf{MTTR} = \frac{1}{\mu}$$

- **MTBF** (**M**ean **T**ime **B**etween **F**ailure):
  („Mittlere Betriebsdauer zwischen zwei Ausfällen")

$$\mathbf{MTBF} = \mathbf{MTTR} + \mathbf{MTTF}$$

Berechnung der Verfügbarkeit



– Berechnung der Verfügbarkeit über:

$$V = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR} = \frac{\mu}{\mu + \lambda}$$

– Ziel ist das erreichen einer hohen Verfügbarkeit: $\frac{MTTR}{MTTF} \ll 1 \rightarrow V \cong 1$

Verfügbarkeit	Ausfallzeit / Jahr	Systemtyp
90,0%	ca. 880h	Unmanaged
99,0%	ca. 88h	Managed (*Standard für Systeme*)
99,9%	ca. 9h	Well-Managed
99,99%	ca. 53min	Fault-Tolerant
99,999%	ca. 5min	Highly Available
99,9999%	ca. 32s	Very Highly Available
99,99999%	ca. 3s	Ultra Highly Available (*nur für Mikro-Bauteile, keine Systeme*)

Hochverfügbarkeit (*high availability*)

- Definition nach IEEE:

  “High Availability (HA for short) refers to the availability of resources in a computer system, in the wake of component failures in the system.”

- System ist hochverfügbar, wenn es auch im Fehlerfall weiter verfügbar ist und ohne unmittelbaren menschlichen Eingriff weiter genutzt werden kann
- Erreichbar durch:
  - Hohe Verfügbarkeit durch hohe Zuverlässigkeit:
    - Hoch zuverlässige Komponenten
    - Redundante Strukturen
  - Hohe Verfügbarkeit durch kurze Reparaturzeiten:
    - Kurze Fehlerdiagnosezeit durch Selbstdiagnose-Software
    - Modulare Struktur zum einfachen Austausch von Komponenten

Hochverfügbarkeit nach der Harvard Research Group (HRG)

– Einteilung der Hochverfügbarkeit in AEC-Stufen („*Availability Environment Classification*“)

AEC	Erklärung
0	Funktion kann unterbrochen werden, Datenintegrität ist nicht essentiell.
1	Funktion kann unterbrochen werden, Datenintegrität muss jedoch gewährleistet sein.
2	Funktion darf nur innerhalb festgelegter Zeiten oder zur Hauptbetriebszeit minimal unterbrochen werden.
3	Funktion muss innerhalb festgelegter Zeiten oder während der Hauptbetriebszeit ununterbrochen aufrechterhalten werden.
4	Funktion muss ununterbrochen aufrechterhalten werden, 24/7-Betrieb muss gewährleistet sein.
5	Funktion muss unter allen Umständen verfügbar sein.

– Probleme beim Erreichen einer hohen Verfügbarkeit:

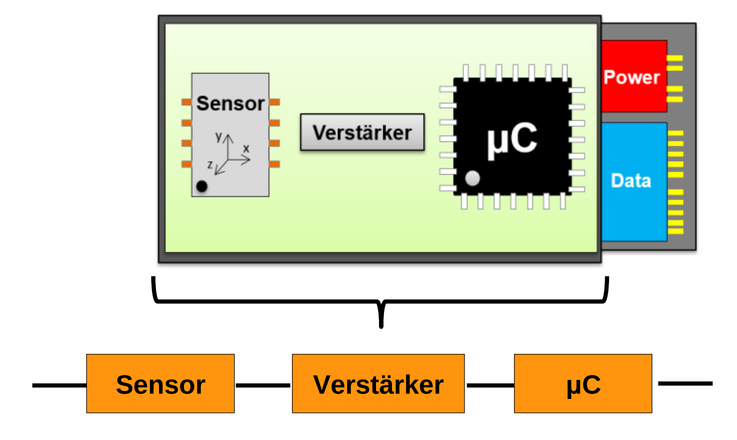
Frage zu Kapitel 2.3

Welche Verfügbarkeit weist eine Anlagensteuerung mit einer Ausfallrate von $\lambda_0 = 6{,}5 \cdot 10^{-4}\ h^{-1}$ auf, wenn die durchschnittliche Reparaturzeit 20 h beträgt?

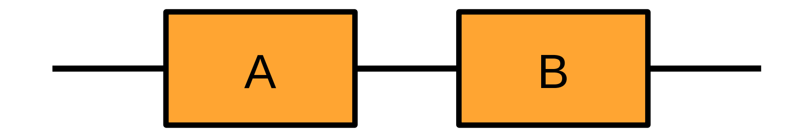
§ 2 Wahrscheinlichkeit und Zuverlässigkeit

Zuverlässigkeitsblockdiagramm

- Betrachtung eines aus mehreren Komponenten aufgebauten Gesamtsystems
- Zuverlässigkeitsblockdiagramm stellt dar, unter welcher Bedingung das Gesamtsystem funktionsfähig ist
- Gesamtsystem ist funktionsfähig, wenn das Blockdiagramm auf mindestens einem Pfad durchlaufen werden kann, auf welchem kein Modul ausgefallen ist
- Block ist eine Komponente bzw. eine bereits zusammengefasste Einheit von Komponenten



Serienanordnung



- System ist funktionsfähig, wenn alle Module funktionsfähig sind
- Sobald mindestens ein Modul ausfällt, ist das System ausgefallen
- Es gilt:
  - Boolesche Funktion: $\mathbf{y = (x_A \wedge x_B)}$
  - Ausfallrate und MTTF:

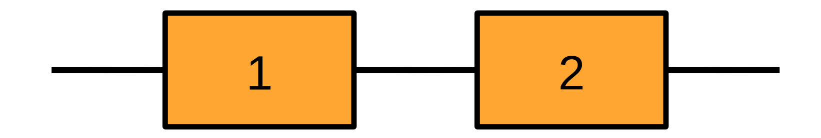
$$\boldsymbol{\lambda_s(t) = \sum_{i=1}^{n} \lambda_i}$$

$$\mathbf{MTTF_s = \frac{1}{\lambda_s(t)} = \frac{1}{\sum_{i=1}^{n} \lambda_i}}$$

  - Zuverlässigkeit:

$$\mathbf{R_s(t) = \prod_{i=1}^{n} R_i(t) = R_1(t) * R_2(t) * \cdots * R_n(t) = e^{-\lambda_1 * t} * e^{-\lambda_2 * t} * \cdots * e^{-\lambda_n * t}}$$

Serienanordnung mit 2 identischen Komponenten



– Für identische Ausfallraten mit λ(t) = const. = $\lambda_1 = \lambda_2 = \lambda_0$ gilt:

- Ausfallrate und MTTF:

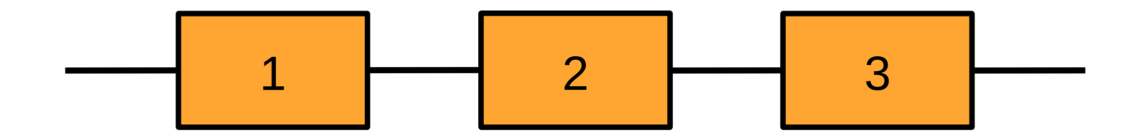
$$\lambda_s(t) = \sum_{i=1}^{2} \lambda_i = \lambda_1 + \lambda_2 \overset{\text{def}}{=} \lambda_0 + \lambda_0 = 2 * \lambda_0$$

$$MTTF_s = \frac{1}{\sum_{i=1}^{2} \lambda_i} = \frac{1}{\lambda_1 + \lambda_2} \overset{\text{def}}{=} \frac{1}{\lambda_0 + \lambda_0} = \frac{1}{2 * \lambda_0}$$

- Zuverlässigkeit:

$$R_s(t) = \prod_{i=1}^{2} R_i(t) = R_1(t) * R_2(t) \overset{\text{def}}{=} R_0(t) * R_0(t) = (R_0(t))^2$$

Serienanordnung mit 3 identischen Komponenten



– Für identische Ausfallraten mit $\lambda(t)$ = const. = $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_0$ gilt:

- Ausfallrate und MTTF:

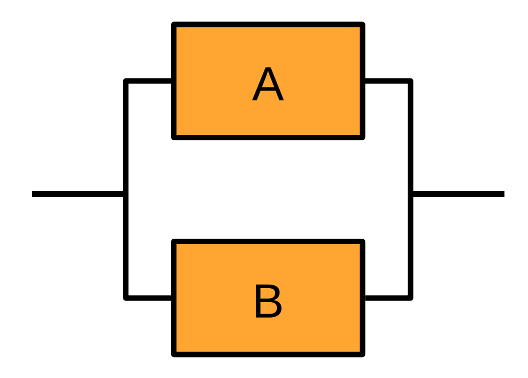
$$\lambda_s(t) = \sum_{i=1}^{3} \lambda_i \stackrel{\text{def}}{=} 3 * \lambda_0$$

$$MTTF_s = \frac{1}{\sum_{i=1}^{3} \lambda_i} \stackrel{\text{def}}{=} \frac{1}{3 * \lambda_0}$$

- Zuverlässigkeit:

$$R_s(t) = \prod_{i=1}^{3} R_i(t) \stackrel{\text{def}}{=} (R_0(t))^3$$

Parallelanordnung (Redundanz)



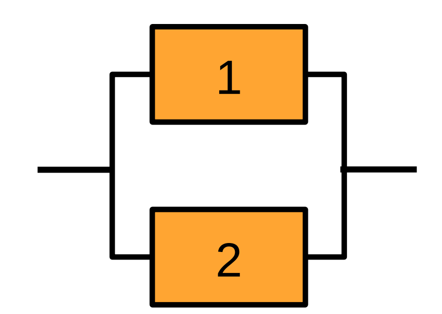
– System ist funktionsfähig, wenn mindestens ein Modul funktionsfähig ist

– Sobald alle Module ausgefallen sind, ist das System ausgefallen

– Es gilt:

- Boolesche Funktion: $\mathbf{y = (x_A \vee x_B)}$
- MTTF: $$\mathbf{MTTF_p = \int_0^{\infty} R_p(t)dt}$$
- Zuverlässigkeit: $$\mathbf{R_p(t) = 1 - \prod_{i=1}^{n} [1 - R_i(t)]}$$

Parallelanordnung mit 2 identischen Komponenten



– Für identische Ausfallraten mit $\lambda(t) = \lambda_1 = \lambda_2 = \lambda_0$ gilt:

- Zuverlässigkeit:

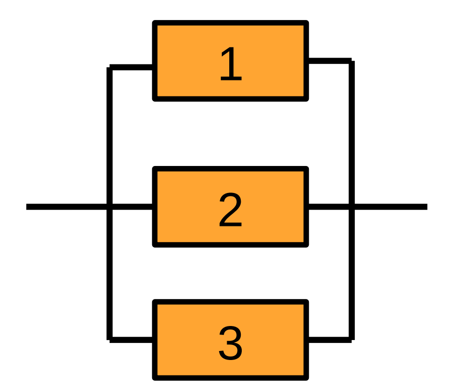
$$\mathbf{R_p(t) = 1 - \prod_{i=1}^{2}[1 - R_i(t)] \stackrel{def}{=} 1 - (1 - R_0(t))^2 = 2R_0(t) - R_0(t)^2}$$

- MTTF:

$$\mathbf{MTTF_p = \int_0^\infty R_p(t)\,dt = \int_0^\infty (2R_0 - {R_0}^2)\,dt = \frac{2}{\lambda_0} - \frac{1}{2\lambda_0} = \frac{1}{\lambda_0} * \left(2 - \frac{1}{2}\right) = \frac{1}{\lambda_0} * \frac{3}{2}}$$

$$\mathbf{MTTF_p = MTTF_0 \sum_{i=0}^{2} \frac{1}{i} = MTTF_0 * \left(1 + \frac{1}{2}\right) = \frac{1}{\lambda_0} * \frac{3}{2}}$$

Parallelanordnung mit 3 identischen Komponenten



- Für identische Ausfallraten mit $\lambda(t) = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_0$ gilt:

  - Zuverlässigkeit:

$$\mathbf{R_p(t) = 1 - \prod_{i=1}^{3}[1 - R_i(t)] \stackrel{def}{=} 1 - (1 - R_0(t))^3 = 3R_0(t) - 3R_0(t)^2 + R_0(t)^3}$$

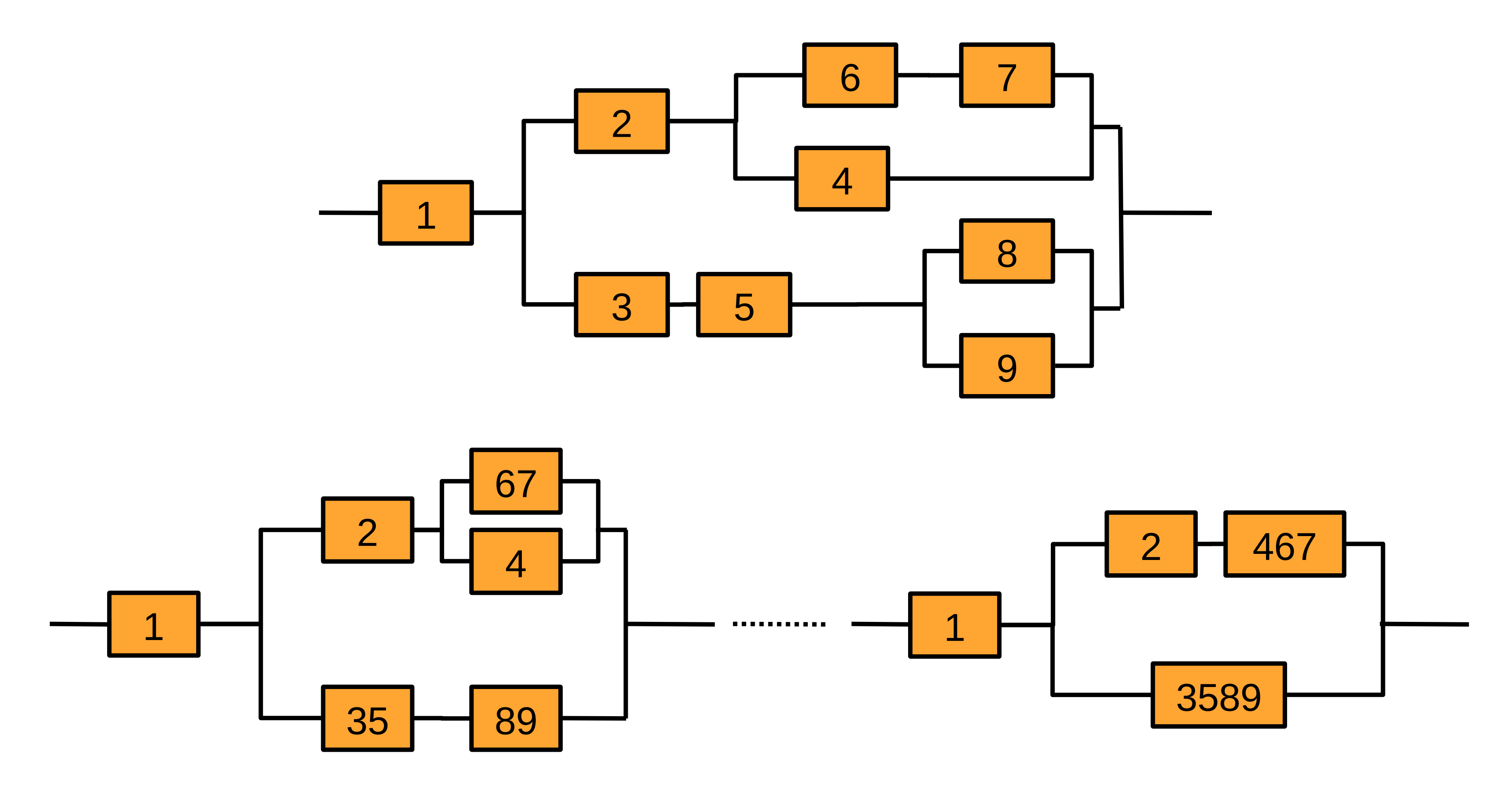
  - MTTF:

$$\mathbf{MTTF_p = \int_0^\infty R_p(t)\,dt = \int_0^\infty (3R_0 - 3R_0{}^2 + R_0{}^3)\,dt = \frac{3}{\lambda_0} - \frac{3}{2\lambda_0} + \frac{1}{3\lambda_0} = \frac{1}{\lambda_0} * \frac{11}{6}}$$

$$\mathbf{MTTF_p = MTTF_0 \sum_{i=0}^{3} \frac{1}{i} = MTTF_0 * \left(1 + \frac{1}{2} + \frac{1}{3}\right) = \frac{1}{\lambda_0} * \frac{11}{6}}$$

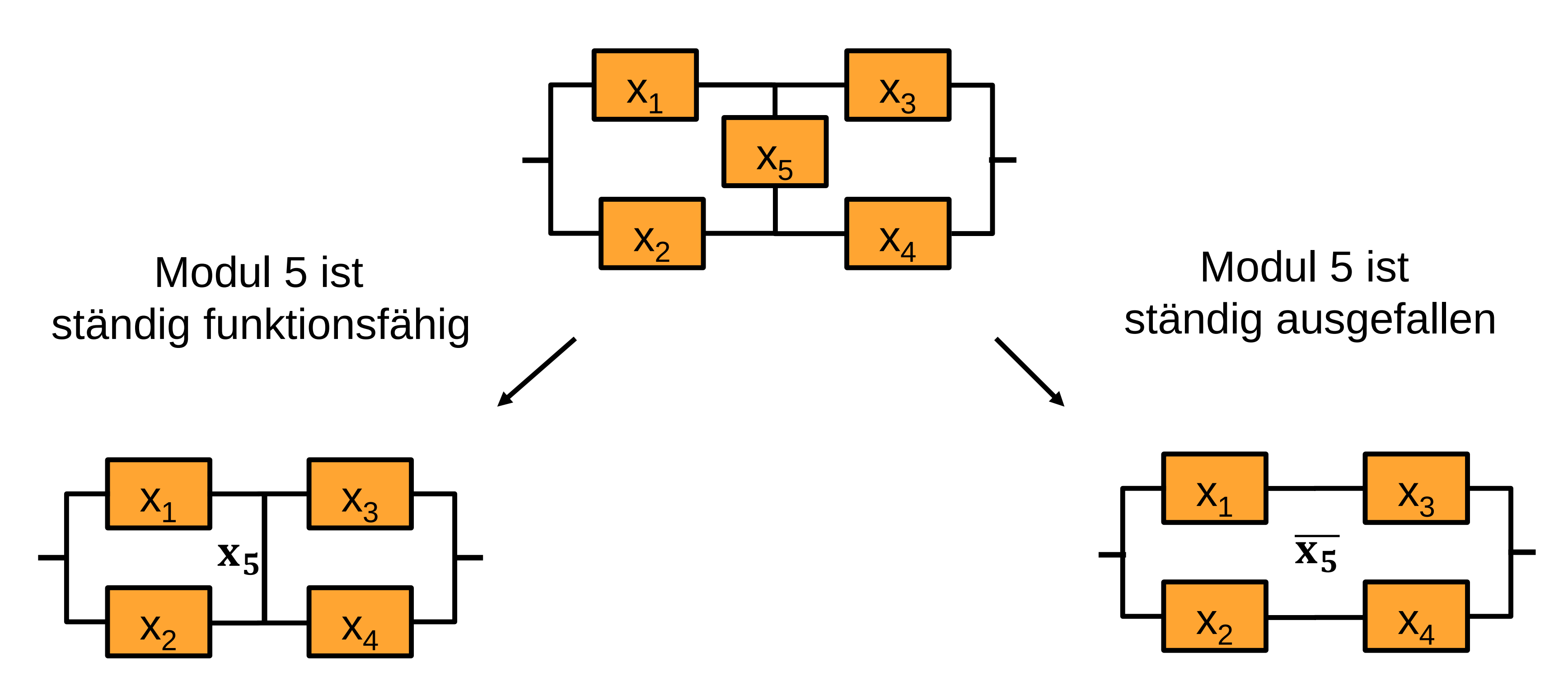
Komplexe Zuverlässigkeitsblockschaltbilder

– Reduzierung über Serien- und Parallelanordnungen:



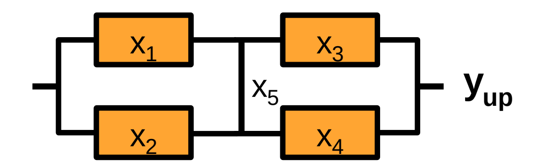
Separation bei Brückenelementen

– Bei Brückenelementen ist eine allgemeine Reduktion auf Serien- bzw. Parallelanordnungen nicht möglich, daher Durchführung einer Separation über Methode der relevanten Systemkomponenten:



Methode der relevanten Systemkomponenten (1/2)

– Betrachtung des Falls „R_5 ist ständig funktionsfähig“:



- Boolesche Funktion:

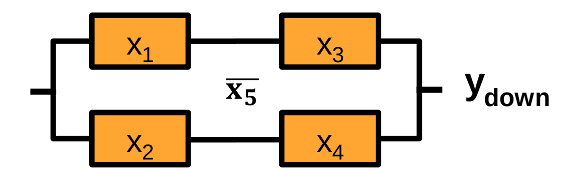
$$\mathbf{y_{up}} = \mathbf{x_5} \wedge [(\mathbf{x_1} \wedge \mathbf{x_3}) \vee (\mathbf{x_1} \wedge \mathbf{x_4}) \vee (\mathbf{x_2} \wedge \mathbf{x_3}) \vee (\mathbf{x_2} \wedge \mathbf{x_4})]$$

- Umformung
  - Distributivgesetz: $\mathbf{y_{up}} = \mathbf{x_5} \wedge [(\mathbf{x_1} \wedge (\mathbf{x_3} \vee \mathbf{x_4})) \vee (\mathbf{x_2} \wedge (\mathbf{x_3} \vee \mathbf{x_4}))]$
  - Kommutativgesetz: $\mathbf{y_{up}} = \mathbf{x_5} \wedge [((\mathbf{x_3} \vee \mathbf{x_4}) \wedge \mathbf{x_1}) \vee ((\mathbf{x_3} \vee \mathbf{x_4}) \wedge \mathbf{x_2})]$
  - Distributivgesetz: $\mathbf{y_{up}} = \mathbf{x_5} \wedge [(\mathbf{x_3} \vee \mathbf{x_4}) \wedge (\mathbf{x_1} \vee \mathbf{x_2})]$

- Zuverlässigkeit:

$$\mathbf{R_{up}} = \mathbf{R_5} * \left[\left(\left(\mathbf{1} - (\mathbf{1} - \mathbf{R_3}) * (\mathbf{1} - \mathbf{R_4})\right) * \left(\mathbf{1} - (\mathbf{1} - \mathbf{R_1}) * (\mathbf{1} - \mathbf{R_2})\right)\right)\right]$$

Methode der relevanten Systemkomponenten (2/2)



– Betrachtung des Falls „R_5 ist ständig ausgefallen“:

- Boolesche Funktion:

$$\mathbf{y_{down}} = \overline{\mathbf{x_5}} \wedge [(\mathbf{x_1} \wedge \mathbf{x_3}) \vee (\mathbf{x_2} \wedge \mathbf{x_4})]$$

- Zuverlässigkeit

$$\mathbf{R_{down}} = (\mathbf{1} - \mathbf{R_5}) * \left[\left(\mathbf{1} - (\mathbf{1} - \mathbf{R_1 R_3}) * (\mathbf{1} - \mathbf{R_2 R_4})\right)\right]$$

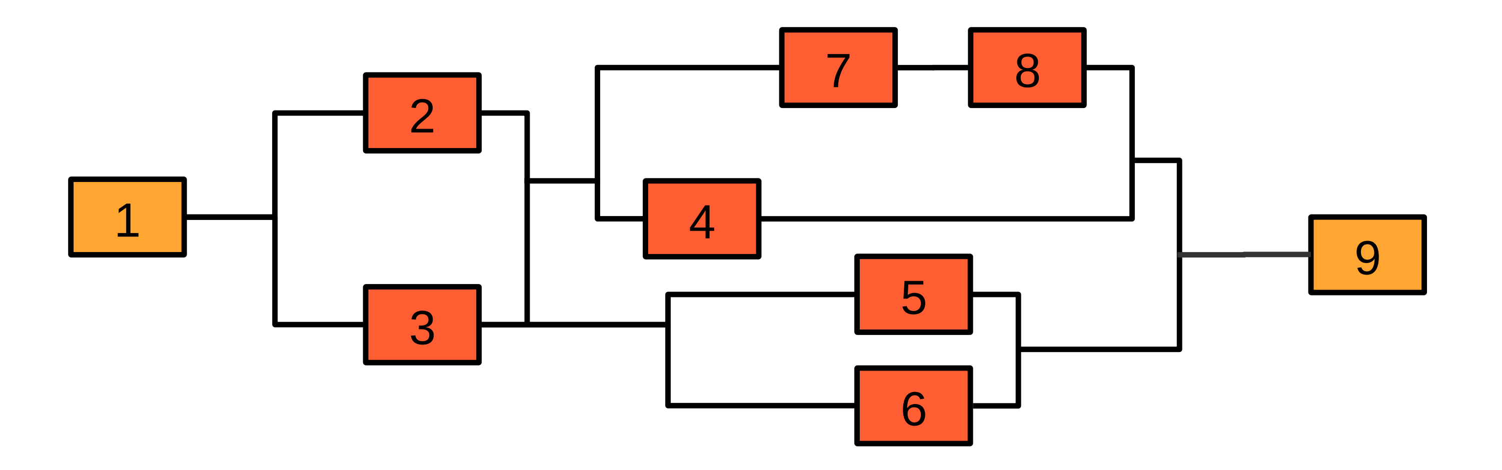
– Berechnung der Systemzuverlässigkeit des Brückenelements über die Addition der Teilwahrscheinlichkeiten

– Möglich, da Ereignisse voneinander unabhängig sind (vgl. dazu Satz der totalen Wahrscheinlichkeit)

– Es gilt also für ein Brückenelement:

Obere Grenze der Zuverlässigkeit

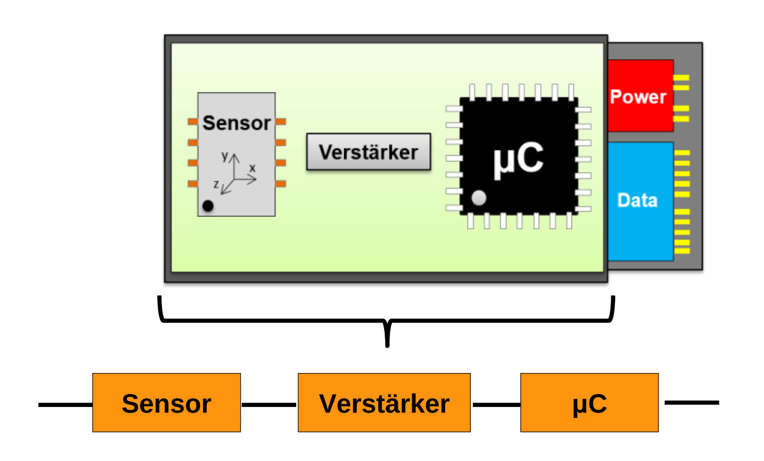
– Bei komplexen Systemen kann die obere Grenze bzw. die maximale Zuverlässigkeit schnell abgeschätzt werden:



– Abschätzung max. Zuverlässigkeit des Systems:

Zuverlässigkeitsanalyse über Blockdiagramme (1/2)

– Am Beispiel eines Steuergeräts:



$$\lambda_{Sensor} = 2 * 10^{-6}\,h^{-1}$$

$$\lambda_{Verst.} = 1 * 10^{-9}\,h^{-1}$$

$$\lambda_{\mu C} = 0,5 * 10^{-6}\,h^{-1}$$

$$t = 10a = 87.600\,h$$

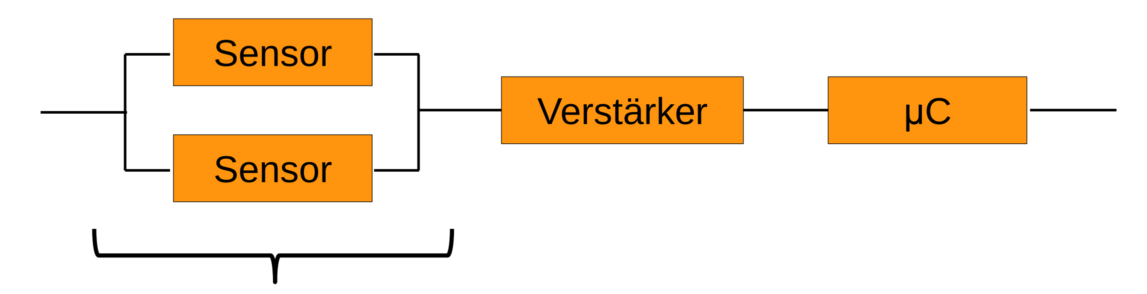
$$R_{Sensor}(87.600h) = e^{-(2*10^{-6}h^{-1}*87600h)} = 0,839$$

$$R_{Verst.}(87.600h) = e^{-(1*10^{-9}h^{-1}*87600h)} = 0,999$$

$$R_{\mu C}(87.600h) = e^{-(0,5*10^{-6}h^{-1}*87600h)} = 0,957$$

Zuverlässigkeitsanalyse über Blockdiagramme (2/2)

– Maßnahme: Redundanz zur Erhöhung der Zuverlässigkeit



$$\mathbf{R_{Sensor}^{(p)}(t) = 2R_{Sensor}(t) - R_{Sensor}(t)^{\,2} = 2 * 0,839 - (0,839)^2 = 0,974}$$

– Zuverlässigkeit des Steuergeräts **mit** Redundanz:

$$\mathbf{R_{ges,neu}(t) = R_{Sensor}^{(p)}(t) * R_{Verst.}(t) * R_{\mu C}(t) = 0,974 * 0,999 * 0,957 = 0,931}$$

– Vgl. Zuverlässigkeit des Steuergeräts **ohne** Redundanz:

$$\mathbf{R_{ges,alt}(t) = R_{Sensor}(t) * R_{Verst.}(t) * R_{\mu C}(t) = 0,839 * 0,999 * 0,957 = 0,802}$$

Frage zu Kapitel 2.4

Um eine höhere Zuverlässigkeit zu erreichen, soll die Sensoreinheit eines Steuergeräts aus 3 identischen, parallel geschalteten Sensoren aufgebaut werden. Es soll dabei im Durchschnitt höchstens alle 20 Jahre zu einem Ausfall kommen.

Können dazu Sensoren eingesetzt werden, die eine Ausfallrate von $\lambda_0 = 9{,}3 * 10^{-6}\ h^{-1}$ aufweisen?

Gliederung

§ 3 Fehlerbaumanalyse (FTA)

Einführung in die FTA

- Fehlerbaumanalyse (FTA) ist eine Methode, die einen graphischen Zusammenhang zwischen einem Top-Ereignis (Systemausfall, Gefährdung,…) und den Ursachen, die zu diesem Top-Ereignis führen, darstellt
- Deduktives Verfahren, d.h., ausgehend von dem unerwünschten Top-Ereignis werden die möglichen Ursachen gesucht
- Ursachen können dabei entweder alleine oder in Kombination mit anderen Ursachen auftreten und zu einem definierten Fehler führen

Historie (1960er – 1980er)

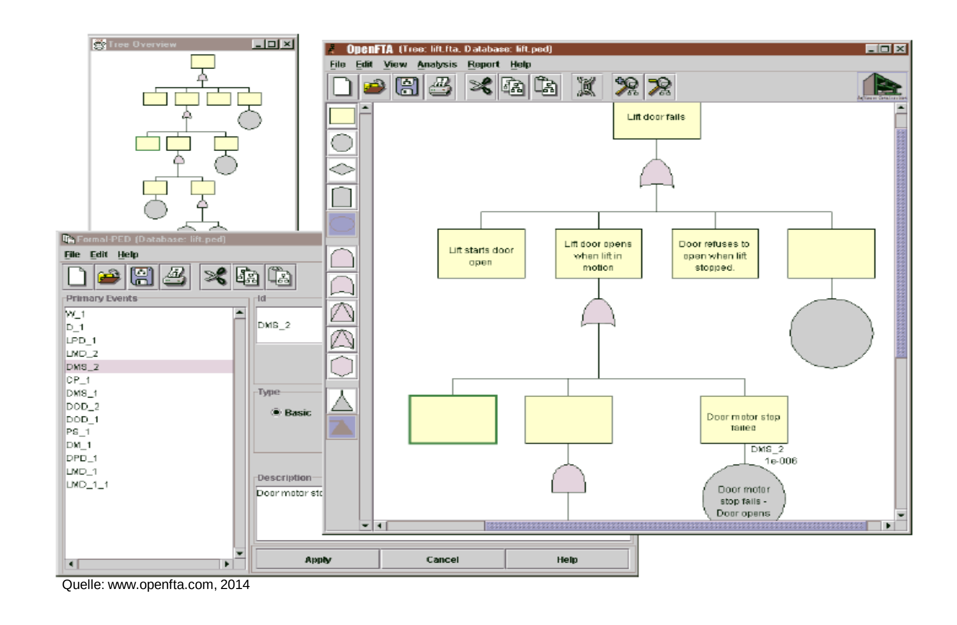

Quelle: en.wikipedia.org, 2014

- 1960er:
  - Entwicklung FTA von H.A. Watson bei Bell Laboratories in New Jersey (USA)
  - Einsatz in der Luft und Raumfahrt u.a. für Flugzeuge von Boeing
- 1970er und 1980er
  - Planung von Kernkraftwerken führte zur internationalen Verbreitung
  - Entstehung Auswerte-Algorithmen und unterstützender Software

Historie (1990er – heute)

– 1990er

- Einsatz in der Automobilindustrie
- Software-Tools zur Konstruktion und Auswertung der FTA



Quelle: www.openfta.com, 2014

– Heute

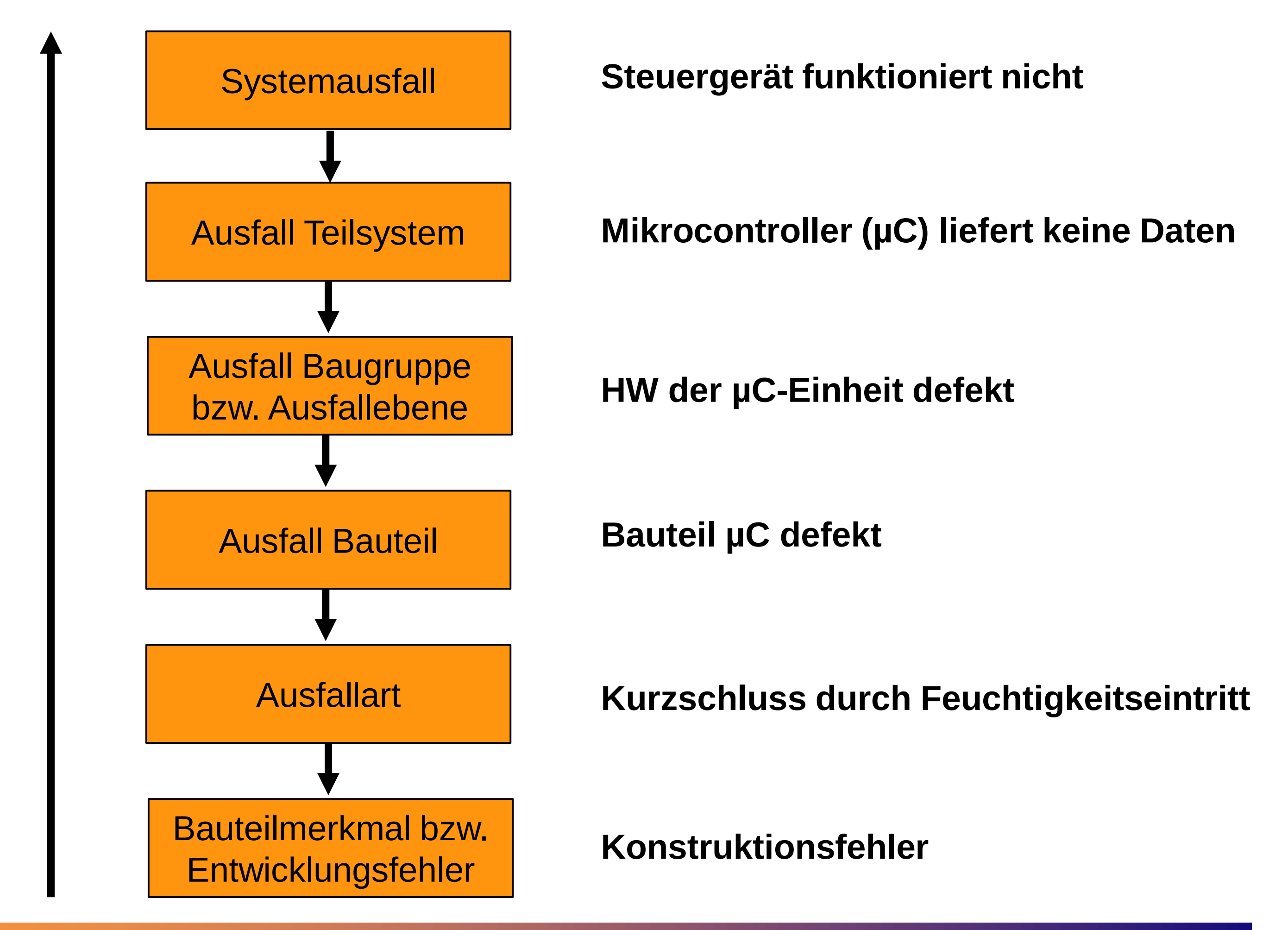
- Anwendung in nahezu allen Bereichen
- Verbreitung der FTA in der SW-Entwicklung
- Normung nach DIN 25424-1/-2 (1981/1990) und DIN EN 61025 (2007)

Ziele der FTA

- System auf Komponentenbasis wirklichkeitsnah zu modellieren und auszuwerten:
  - Ausfallarten und Ausfallursachen zu detektieren
  - Funktionale Zusammenhänge der Ausfälle herzustellen
  - Auswirkungen von Ausfällen auf das System zu beschreiben
- Die FTA wird eingesetzt zur:
  - Präventiven Qualitätssicherung
  - Systemanalyse
  - Problemlösung bei neu auftretenden Fehlern
- Analyse der Zuverlässigkeit über qualitativen oder quantitativen Ansatz:
  - Qualitativ:
  - Quantitativ:

Struktur eines Fehlerbaums (Top-Down)

– Graphische Darstellung über mehrere Systemebenen, die durch logische Verknüpfungen miteinander verbunden sind:



Gängige Symbole des Fehlerbaums (1/2)

- Konstruktion des Fehlerbaums nach DIN 25424-1 genormten Symbolen
- Unterscheidung in Ereignis-, Verknüpfungs- und Tranfersymbole
- Ereignissymbole:

Symbol	Bezeichnung	Erklärung
	Ereignis	Fehlereignis, das aus der Interaktion mehrerer Ereignisse durch logische Verknüpfung resultiert.
	Primäres Ereignis	Feinste Auflösung des Fehlerbaums, z.B., der Ausfall einer Komponente. Wird nicht weiter untergliedert.
	Unentwickeltes Ereignis	Sonderfall, ein Ereignis über das keine weiteren Details bekannt sind. Keine weitere Untergliederung möglich.
	Auslösendes Ereignis	Keine eigenständige Fehlerquelle, tritt als Rahmenbedingung im Betrieb auf. Kombination mit weiteren Ereignissen nötig.

Gängige Symbole des Fehlerbaums (2/2)

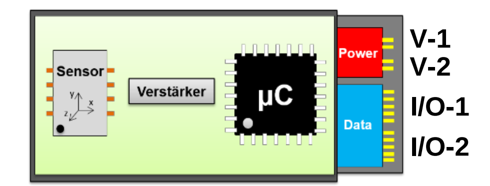
– Verknüpfungssymbole:

Symbol	Bezeichnung	Erklärung
&	UND-Verknüpfung	Ausgangsereignis tritt genau dann ein, wenn alle Eingangsereignisse wahr sind.
≥1	ODER-Verknüpfung	Ausgangsereignis tritt genau dann ein, wenn mindestens ein Eingangsereignis wahr ist.

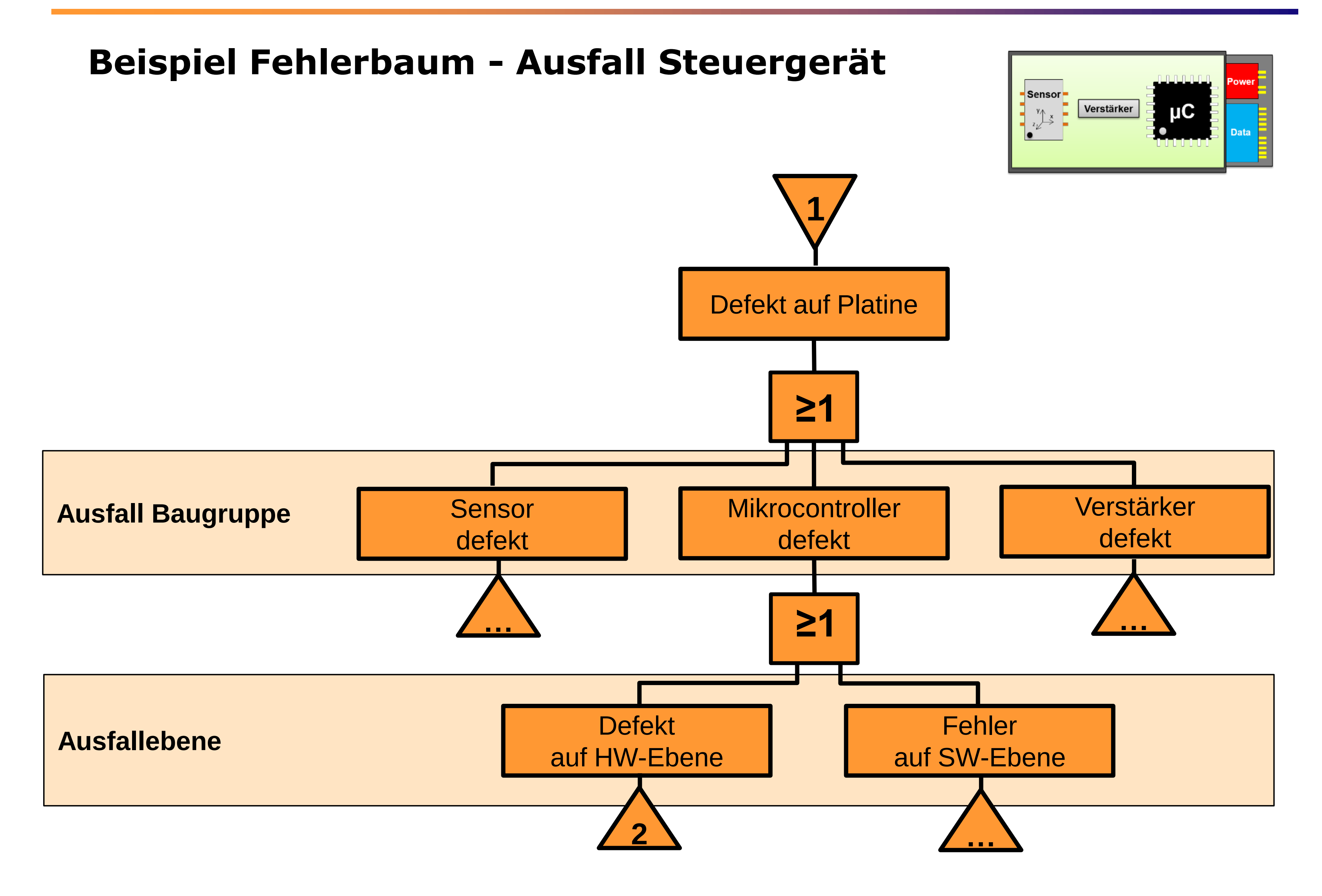
– Tranfersymbole:

Symbol	Bezeichnung	Erklärung
x	Transfer-IN	Input, Ereigniseingang einer Verbindung zu einem anderen Fehlerbaum.
x	Transfer-OUT	Ereignis-Output, der an einen anderen Baum den Input liefert. An Position des Top-Ereignisses.

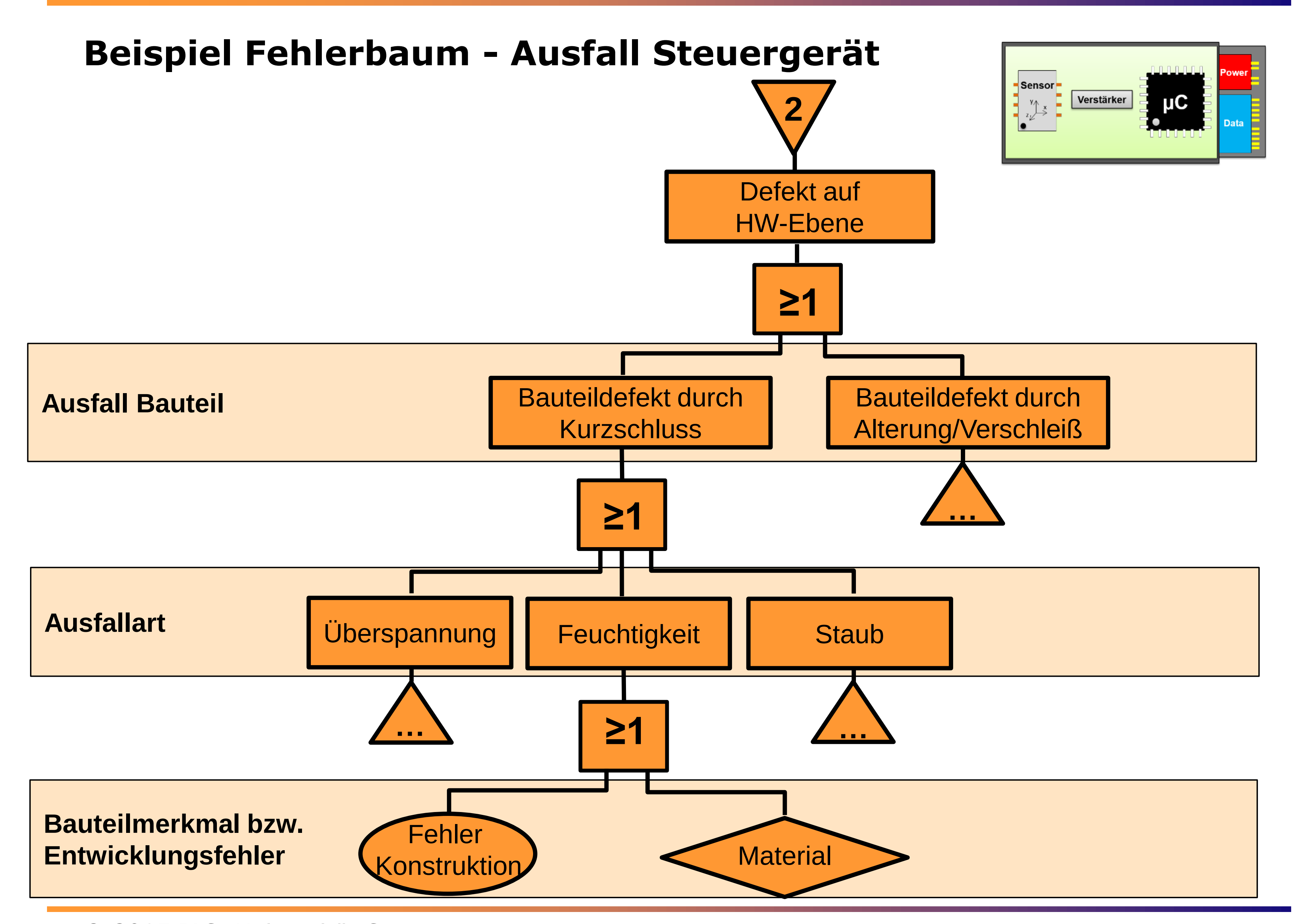
Beispiel Fehlerbaum - Ausfall Steuergerät



Beispiel Fehlerbaum - Ausfall Steuergerät



Beispiel Fehlerbaum - Ausfall Steuergerät



Fragen zu Kapitel 3.1

Welchen Aussagen stimmen Sie zu?

☐ Bei der FTA handelt es sich um ein deduktives Verfahren.

☐ Das Vorgehen der FTA entspricht einer Bottom-Up-Analyse.

☐ Ein primäres Ereignis kann mehrere Fehlerursachen haben.

☐ Das Ergebnis einer FTA kann immer über Zahlenwerte beschrieben werden.

§ 3 Fehlerbaumanalyse (FTA)

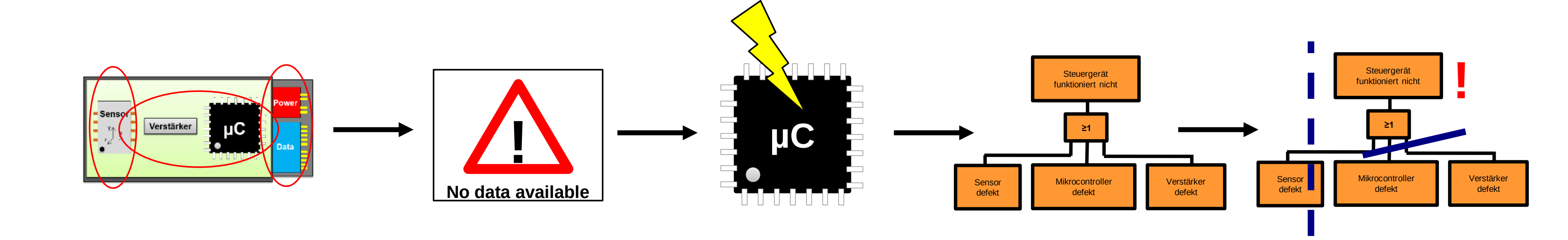
Qualitative FTA – Anwendung (1/2)

– Ziele:
  - Identifikation sämtlicher Ausfallarten und Ausfallursachen
  - Festhalten aller kritischer Ereignisse und Ereigniskombinationen
  - Lokalisierung von Schwachstellen
  - Erstellung objektiver Beurteilungskriterien
  - Dokumentation

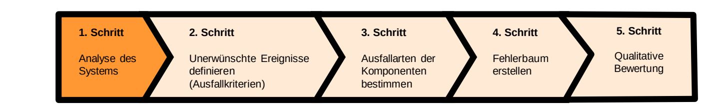
Qualitative FTA – Anwendung (2/2)

– Gliederung:





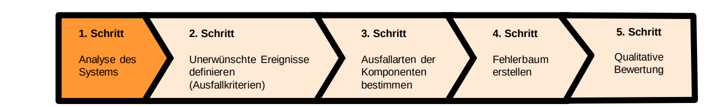
1. Analyse des Systems (1/2)



– Mit dem Ziel:
  - Wirkungsweise des Systems deutlich zu machen
  - Wechselwirkung über Schnittstellen mit der Umwelt zu ermitteln
  - Abhängigkeiten innerhalb des Systems zu ermitteln

– Wirkungsweise des Systems über Zusammenhang zwischen Anforderungen und Funktionen bestimmen:
  - Darlegen der Anforderungen (Funktion, Leistung, Qualität,…)
  - Aufzeigen der Funktionalität und Zuordnung einzelner Funktionen zu Komponenten
  - Durchgängigkeit zwischen Anforderungen, Funktionen und Komponente herstellen

1. Analyse des Systems (2/2)



– Berücksichtigung der Umgebung
  - Identifikation relevanter Umgebungseinflüsse (Temperatur, Feuchtigkeit, Vibration,...)
  - Eigenschaften der Systemelemente bestimmen (chemisch und physikalisch), die von der Umgebung beeinflusst werden können

– Verhaltensabhängigkeiten untersuchen
  - Zusammenwirken von Systemkomponenten
  - Reaktion auf Einflüsse der Umgebung und auf Ausfälle externer/interner, mit der System in Verbindung stehender, Komponenten

2. Unerwünschte Ereignisse definieren



– Vorgehen nach zwei mögliche Ansätze

- Präventiver Ansatz:

  „Unerwünschten Ereignisse werden von der möglichen Nichterfüllung von Funktionen und Anforderungen abgeleitet.“

- Korrektiver Ansatz:

  „Unerwünschte Ereignisse werden direkt von aufgetretenen Ausfällen bzw. Fehlfunktionen abgeleitet.“

3. Ausfallarten bestimmen



- Unterschiedliche Arten von Ausfällen haben unterschiedliche Auswirkungen auf das Top-Event
- Betrachtung verschiedener Arten von Komponentenausfällen:
  - Primärer Ausfall
    - Komponentenausfall durch Schwächen bzw. Fehler, die sich schon zu Beginn im System befinden
    - Ausfall bei zulässigen Einsatzbedingungen
  - Sekundärer Ausfall
    - Komponentenausfall, der durch die Umgebungsbedingungen oder durch die Einsatzbedingungen eintritt
    - Ausfall bei unzulässigen Einsatzbedingungen

3. Ausfallarten bestimmen

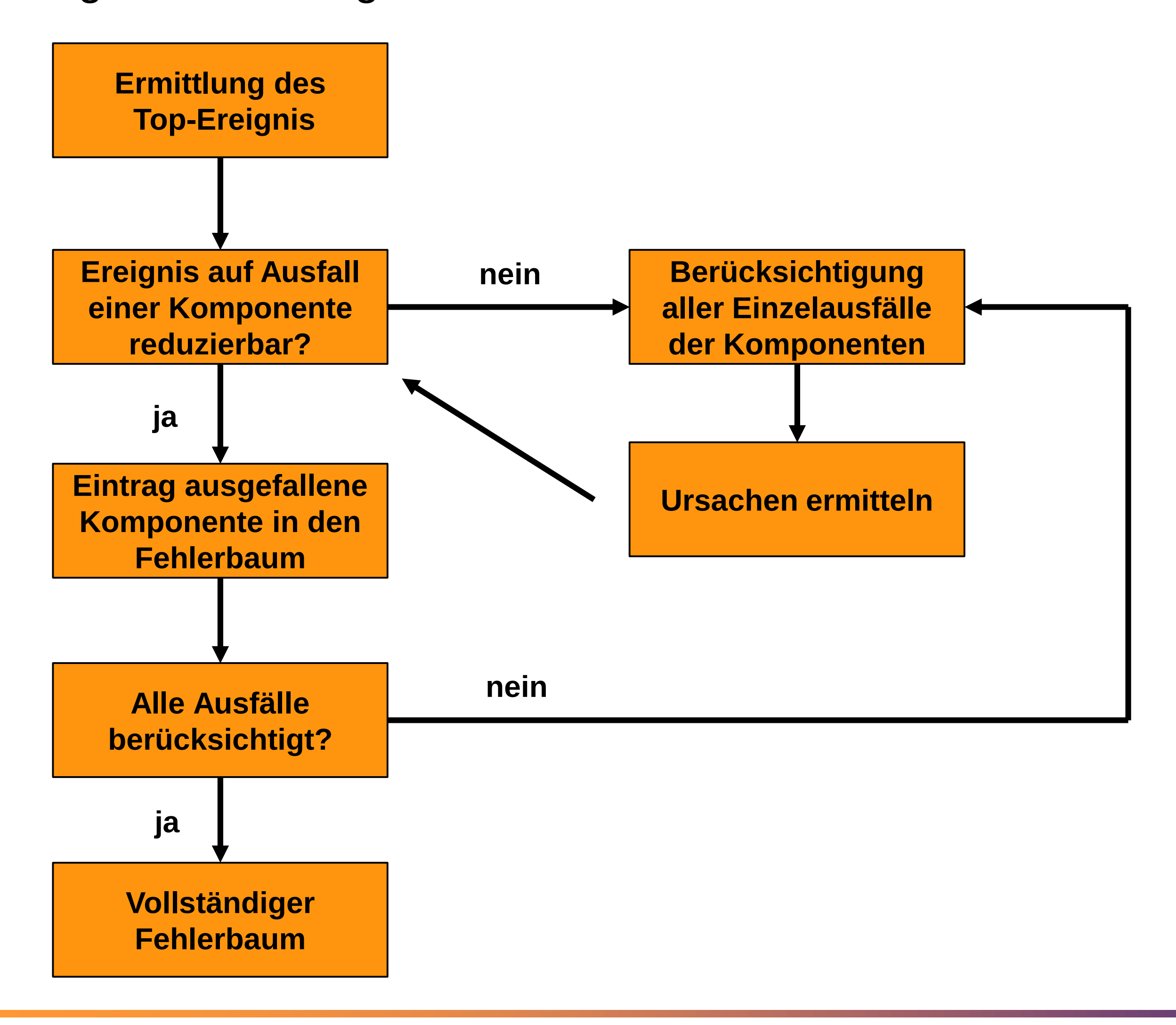


- Kommandierter Ausfall
  - Komponentenausfall einer eigentlich funktionsfähigen Komponente durch eine Fehlbedienung oder Eingabe von falschen oder ungültigen Werten (bei Software)
  - Bedienung- und Wartungsfehler
  - Absichtliche Fehler

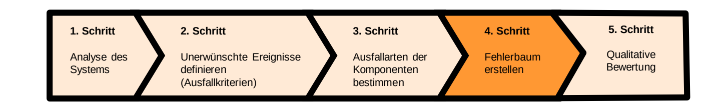
4. Fehlerbaum erstellen

1. Schritt	2. Schritt	3. Schritt	4. Schritt	5. Schritt
Analyse des Systems	Unerwünschte Ereignisse definieren (Ausfallkriterien)	Ausfallarten der Komponenten bestimmen	Fehlerbaum erstellen	Qualitative Bewertung

– Allgemeines Vorgehensmuster:



Common Mode Failure



- Ausfall mehrerer gleichartiger Komponenten, die zu einem Schadenereignis führen
- Fehler, die nicht durch eine gemeinsame Ursache ausgelöst werden

Ausfall Sensoreinheit

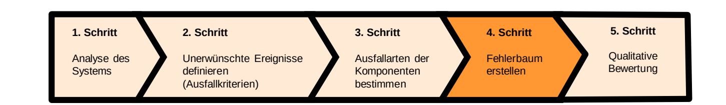
&

Ausfall Sensor 1

Ausfall Sensor 2

Hat der Ausfalle eines Sensors den Ausfall eines anderen Sensors zur Folge?

Common Cause Failure



– Ausfall mehrerer Komponenten, deren Ursache ein einzelner Ausfall einer anderen Komponente darstellt

– Ausfälle sind statistisch abhängig voneinander, da sich verschiedene Äste des Fehlerbaums auf dieselbe ausgefallene Komponente beziehen

...

Ausfall Sensor 1

≥1

Bauteil 1 defekt

Versorgung defekt

...

Ausfall Sensor 2

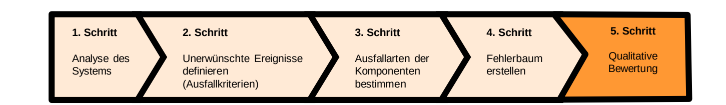
≥1

Bauteil 2 defekt

Versorgung defekt

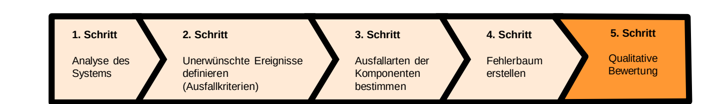
Handelt es sich um denselben Spannungsversorgungsanschluss?

5. Qualitative Bewertung



- Die Zuverlässigkeit bzw. das Risiko von Ausfällen wird über die graphische Struktur (qualitativ) abgeschätzt
- Keine Eingangsdaten für den Fehlerbaum (Ausfallraten) notwendig
- Verfahren bzw. Fehlerbaum muss vollständig sein, damit eine qualitative Bewertung zweckmäßig ist
- Vollständigkeit bedeutet, dass alle Ereignisse und Ereigniskombinationen in der Fehlerbaumerstellung berücksichtig wurden
- Ansätze zu Bewertung:
  - Kritischer Pfad
  - Kritische Menge bzw. minimale Schnittmenge

Kritischer Pfad



- Ast des Fehlerbaums, bei dem die Komponentenausfälle nicht durch systemeigene Vermeidungs- bzw. Prüfmechanismen (Diagnose- und Fehlererkennungsmaßnahmen) abgesichert sind oder werden können
- Erkenntnisse über Zusammenhänge von Ursache und Wirkung
- Ableitung von Näherungen über Risiken und Schwachstellen des Systems

Defekt auf SW-Ebene

≥1

Fehler bei Code-Ausführung

Laufzeitumgebung fehlerhaft

≥1

&

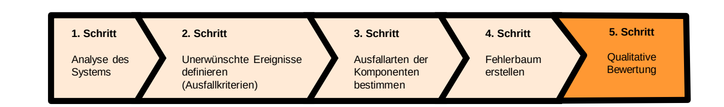
Tatsächlicher SW-Fehler liegt vor

Fälschlicherweise wird ein Fehler gemeldet

Zugriff auf Funktion

Funktion arbeitet anders als erwartet

Kritische Menge – Schwächster Ast



- Die kritische Menge ist der Unterbaum eines Fehlerbaums, der die minimale Kombination von Einzelelementen enthält, deren Ausfall zu einem unerwünschten Ereignis führt
- Es ist somit eine Aussage über den schwächsten Ast des Fehlerbaums möglich
- Schwächster Ast ist die Kombination an Ereignissen, die am ehesten zu einem Ausfall führt

Defekt auf SW-Ebene

≥1

Fehler bei Code-Ausführung

Laufzeitumgebung fehlerhaft

≥1

&

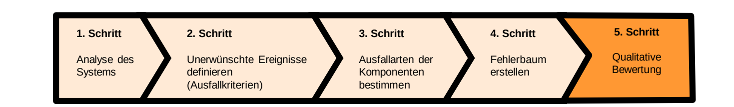
Tatsächlicher SW-Fehler liegt vor

Fälschlicherweise wird ein Fehler gemeldet

Zugriff auf Funktion

Funktion arbeitet anders als erwartet

Kritische Menge – Stärkster Ast



– Ist für einen Unterbaum der schwächste Ast bestimmt worden, so ist es auch möglich, den stärksten Ast des Unterbaums zu bestimmen

– Stärkster Ast ist die Kombination an Ereignissen, mit deren Auftreten am ehesten nicht zu rechnen ist und somit ein Ausfall „unrealistischer“ ist

– D.h., ein Ausfall über den stärksten Ast wird am wenigsten erwartet

Defekt auf SW-Ebene

≥1

Fehler bei Code-Ausführung

Laufzeitumgebung fehlerhaft

≥1

&

Tatsächlicher SW-Fehler liegt vor

Fälschlicherweise wird ein Fehler gemeldet

Zugriff auf Funktion

Funktion arbeitet anders als erwartet

Beurteilung der qualitativen FTA

- Vorteile:
  - Exakte Anpassung an den Untersuchungsgegenstand möglich
  - Tiefer Informationsgehalt der Auswertung
  - Ermöglicht die Aufdeckung noch unbekannter Ausfallursachen

- Nachteile:
  - Aufwendige Auswertung der Ergebnisse
  - Hohe fachliche Kenntnis der Themenfelder der jeweiligen Äste notwendig
  - Relativ zeit- und kostenintensiv

Fragen zu Kapitel 3.2

Welchen Aussagen stimmen Sie zu?

☐ Der erste Schritt einer FTA ist die Erstellung eines Fehlerbaums.

☐ Eine qualitative FTA findet Einsatz bei der Dokumentation.

☐ Eine qualitative Auswertung hat einen tiefen Informationsgehalt.

☐ Qualitative FTA ermöglicht die Vergleichbarkeit von Systemen.

§ 3 Fehlerbaumanalyse (FTA)

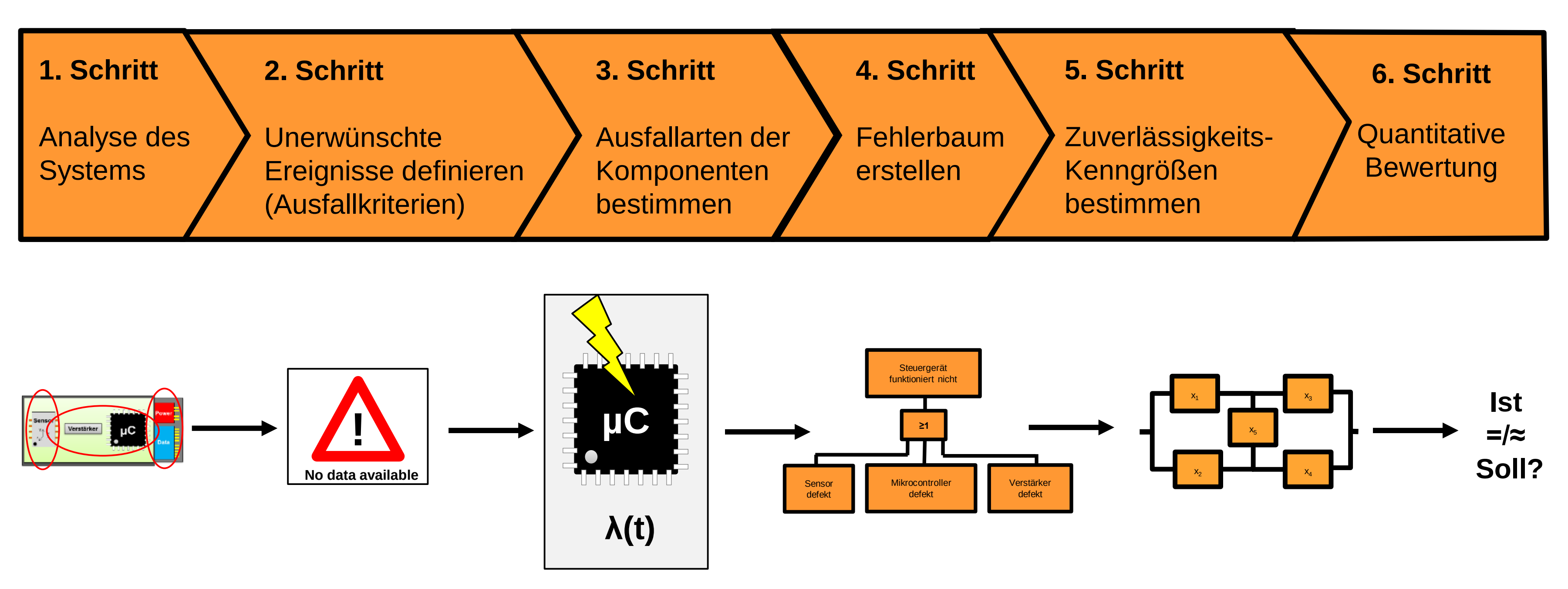
Quantitative FTA – Anwendung (1/2)

– Ziele:

- Nachweis geforderter Zuverlässigkeitsanforderungen
- Berechnung von Zuverlässigkeitskenngrößen bzgl. konkreter Zahlenwerte (Ausfall- und Überlebenswahrscheinlichkeit)
- Aufzeigen von Faktoren oder Komponenten, die die Zuverlässigkeit besonders beeinflussen

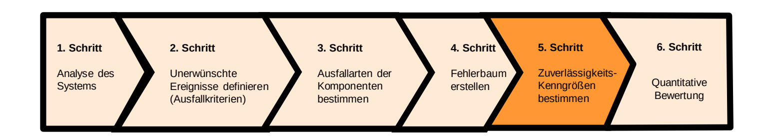
Quantitative FTA – Anwendung (2/2)

– Durchführung:



Zuverlässigkeitskenngrößen der Komponenten benötigt!

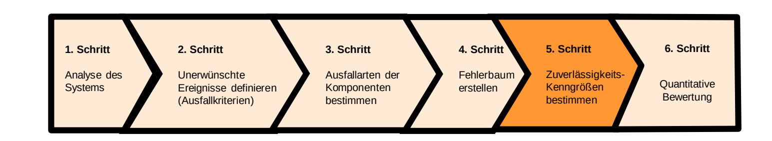
5. Zuverlässigkeitskenngrößen



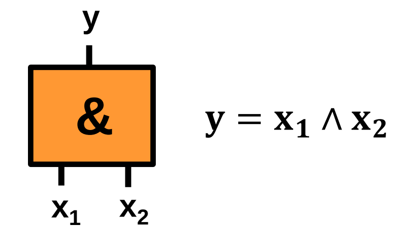
– Zuverlässigkeitskenngrößen der Komponenten (Ausfallrate, Zuverlässigkeit):
  - Bekannt durch Herstellerangaben
  - Ermittelt über Labortests
  - Geschätzt über Wahrscheinlichkeitsrechnung

– Bestimmung der Zuverlässigkeitskenngrößen für Basisereignisse
  - Ausfallwahrscheinlichkeit über einen Zeitraum berechnen, d.h., die Berechnung der zu erwarteten Eintrittshäufigkeit des Top-Events
  - Nichtverfügbarkeit zu einem beliebigen Zeitpunkt berechnen

Funktions- und Fehlerbaum

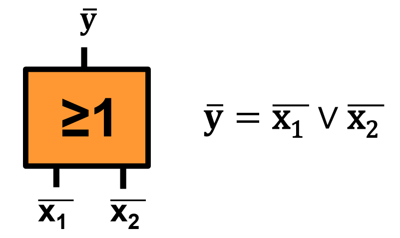


- Es gibt eine grundsätzliche Unterscheidung zwischen einer Funktionsbaum- und einer Fehlerbaum-Betrachtung
- Es gilt für eine Funktionsbaum-Betrachtung, dass Ereignisse bzw. positive Parameter verknüpft werden, zum Beispiel:



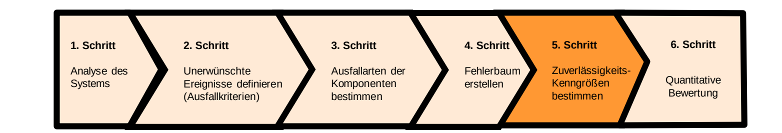
$$\mathbf{y} = \mathbf{x_1} \wedge \mathbf{x_2}$$

- Es gilt für eine Fehlerbaum-Betrachtung, dass Fehl-Ereignisse bzw. negative Parameter verknüpft werden, zum Beispiel :



$$\overline{\mathbf{y}} = \overline{\mathbf{x_1}} \vee \overline{\mathbf{x_2}}$$

Gegenseitige Überführung (1/2)



$R_{ges}(t)$, y, &, x_1, x_2, $R_1(t)$, $R_2(t)$

$$y = x_1 \wedge x_2$$

$F_{ges}(t)$, $\overline{y}$, ≥1, $\overline{x_1}$, $\overline{x_2}$, $F_1(t)$, $F_2(t)$

$$\overline{y} = \overline{x_1} \vee \overline{x_2}$$

$$F_{ges}(t) = 1 - [(1 - F_1(t)) * (1 - F_2(t))]$$

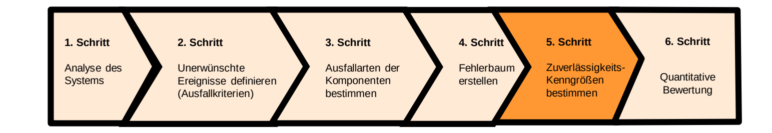
$$1 - R_{ges}(t) = 1 - \Big[\Big(1 - \big(1 - R_1(t)\big)\Big) * \Big(1 - \big(1 - R_2(t)\big)\Big)\Big]$$

$$1 - R_{ges}(t) = 1 - [R_1(t) * R_2(t)]$$

$$R_{ges}(t) = R_1(t) * R_2(t)$$

$$R_{ges}(t) = R_1(t) * R_2(t)$$

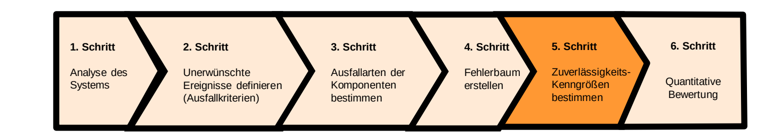
Gegenseitige Überführung (2/2)



– Es sich lässt sich demnach folgender Zusammenhang ableiten:

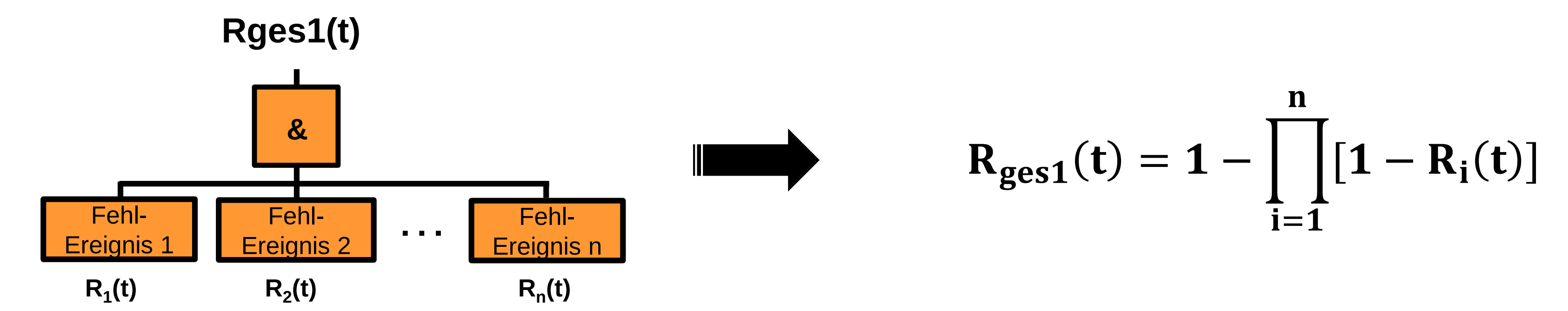
Funktionsbaum	Fehlerbaum	Zuverlässigkeitsberechnung
y & x_1 x_2	$\bar{y}$ ≥1 $\bar{x}_1$ $\bar{x}_2$	
y ≥1 x_1 x_2	$\bar{y}$ & $\bar{x}_1$ $\bar{x}_2$	

Berechnung am Fehlerbaum



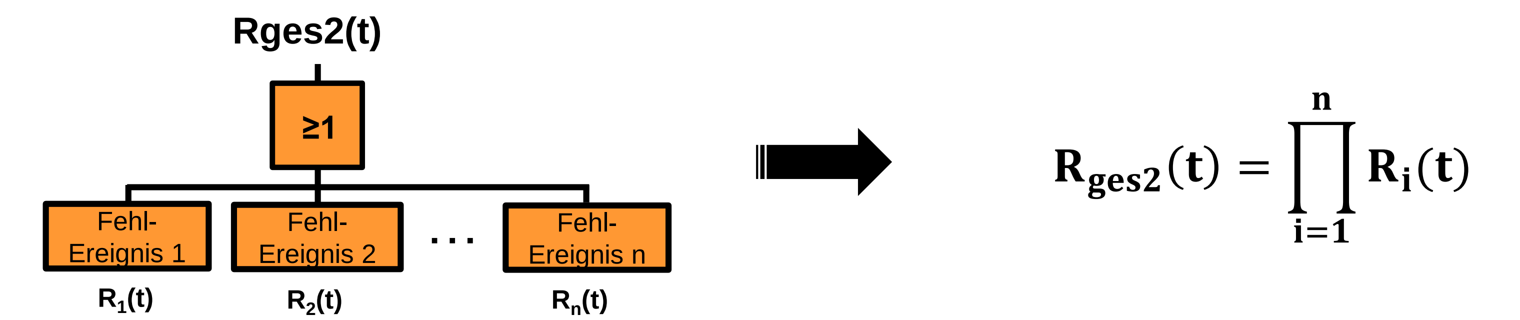
– Die quantitative Zuverlässigkeitsuntersuchung bei Fehlerbäumen lässt sich demnach zu folgendem Schema zusammenfassen:

- UND-Verknüpfungen



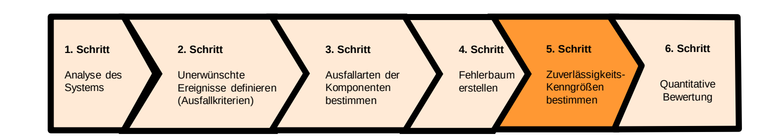
$$R_{ges1}(t) = 1 - \prod_{i=1}^{n} [1 - R_i(t)]$$

- ODER-Verknüpfung



$$R_{ges2}(t) = \prod_{i=1}^{n} R_i(t)$$

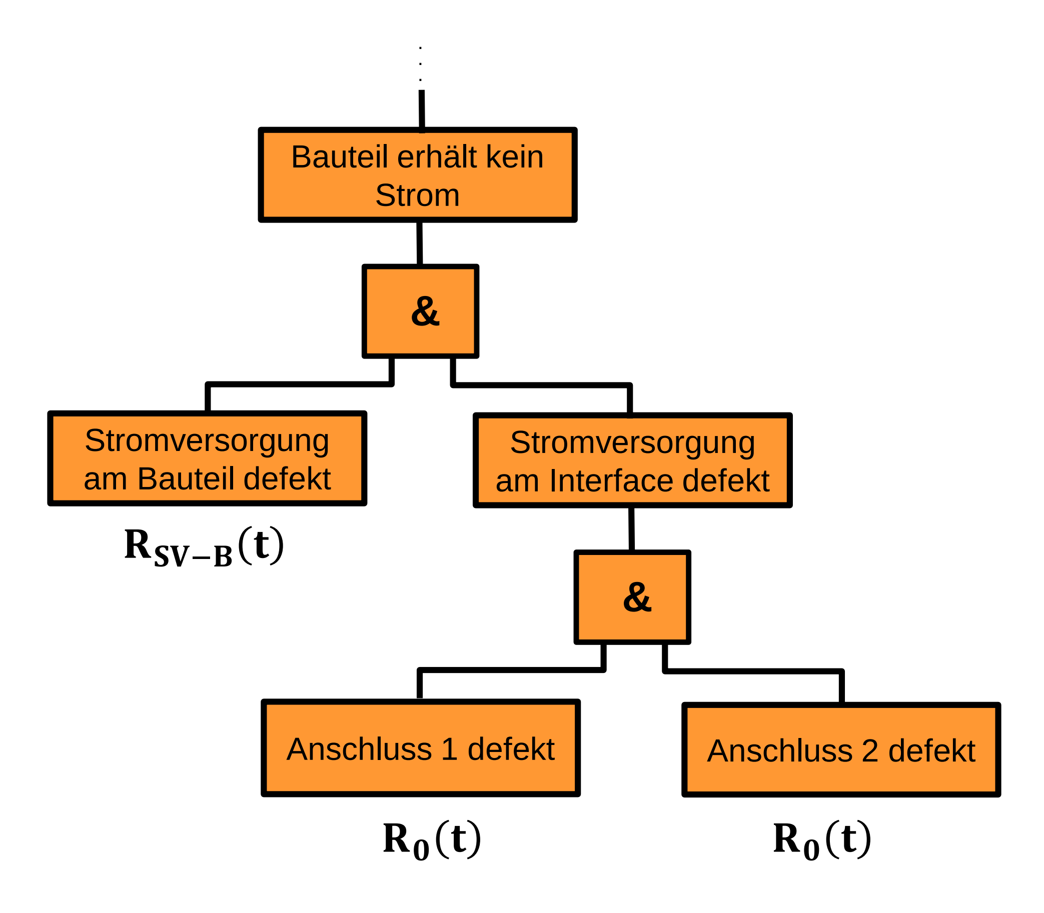
Zusammenfassen von Pfaden



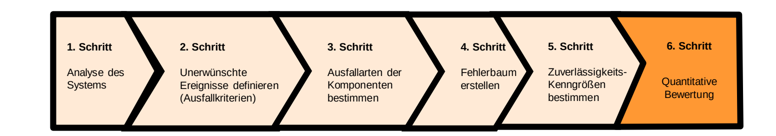
– Um die Analyse größerer Fehlerbäume zu ermöglichen, werden die Zuverlässigkeiten entlang einzelner Pfade nach Schema zusammengefasst

– Beispiel: $\mathbf{t = 10a = 87600h}$

$$\boldsymbol{\lambda_0 = 0{,}5 * 10^{-6}\ h^{-1}}$$

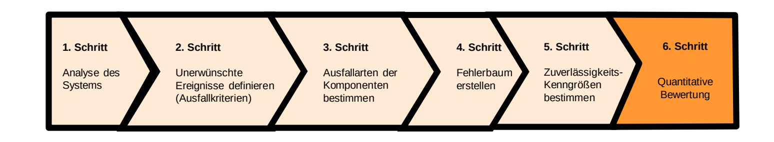


6. Quantitative Bewertung

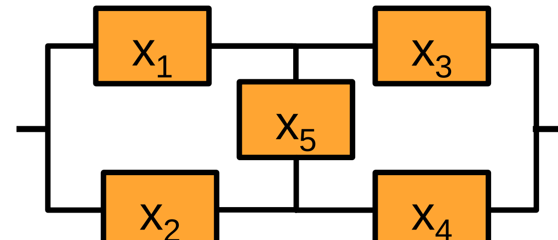


- Bei der Durchführung einer quantitativen Bewertung gilt es, den Ist-Zustand eines Gesamtsystem anhand von charakteristischen und eindeutigen Zuverlässigkeitskenngrößen mit dem geplanten und vorher definierten Soll-Zustand zu vergleichen
- Methoden dabei:
  - Methode der minimalen Erfolgspfade
  - Methode der minimalen Ausfallschnitte

Minimale Erfolgspfade (1/2)



- Ein Pfad entspricht einer Menge an Komponenten, deren Funktionsfähigkeit das Funktionieren des Gesamtsystems garantiert
- Pfade sind dabei minimale Pfade, d.h., ein Pfad darf keine weiteren (Unter-) Pfade als Teilmenge enthalten
- Für jeden Pfad wird die Zuverlässigkeit anhand der darin enthaltenen Komponenten berechnet, zum Beispiel:



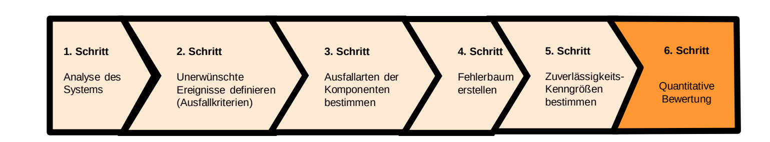
- Pfade als solche stellen Serienanordnungen dar

$$\mathbf{p_1} = (\mathbf{x_1} \wedge \mathbf{x_3}) \qquad \mathbf{p_2} = (\mathbf{x_2} \wedge \mathbf{x_4})$$

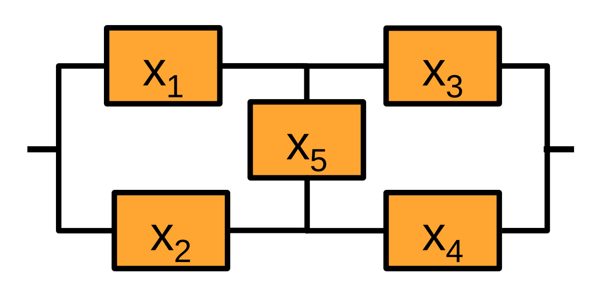
- Menge an Pfaden stellt Parallelanordnung dar

$$\mathbf{p_{12}} = (\mathbf{x_1} \wedge \mathbf{x_3}) \vee (\mathbf{x_2} \wedge \mathbf{x_4})$$

Minimale Erfolgspfade (2/2)



– Beispiel:



- Pfad 1: $\mathbf{p_1 = (x_1 \wedge x_3)}$
- Pfad 2: $\mathbf{p_2 = (x_2 \wedge x_4)}$
- Pfad 3: $\mathbf{p_3 = (x_1 \wedge x_5 \wedge x_4)}$
- Pfad 4: $\mathbf{p_4 = (x_2 \wedge x_5 \wedge x_3)}$

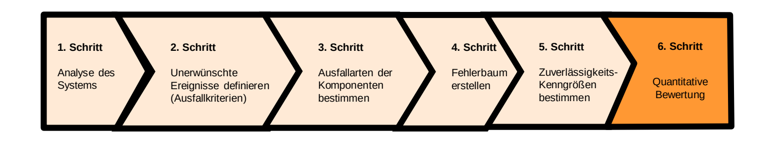
– Boolesche Verknüpfung der Pfade:

$$\mathbf{p_{ges} = (x_1 \wedge x_3) \vee (x_2 \wedge x_4) \vee (x_1 \wedge x_5 \wedge x_4) \vee (x_2 \wedge x_5 \wedge x_3)}$$

– Berechnung der Systemzuverlässigkeit (boolesche Modellbildung):

$$\mathbf{R_{ges} = \left[1 - \left((1 - R_1R_3) * (1 - R_2R_4) * (1 - R_1R_5R_4) * (1 - R_2R_5R_3)\right)\right]}$$

Minimale Ausfallschnitte (1/2)



- Ein Schnitt ist der komplementäre Begriff zu einem Pfad, d.h., ein Schnitt ist die Menge an Komponenten, deren Ausfall zum Ausfall des Gesamtsystems führt
- Schnitte sind dabei minimale Schnitte, d.h., ein Schnitt darf keine weiteren Schnitte als Teilmenge enthalten
- Für jeden Schnitt wird die Ausfallwahrscheinlichkeit anhand der darin enthaltenen Komponenten berechnet, zum Beispiel:
- Schnitte als solche stellen Serienanordnungen dar

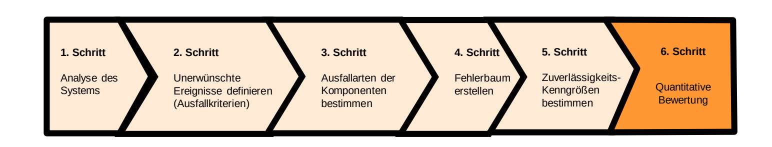
$\mathbf{c_1} = (\overline{\mathbf{x_1}} \wedge \overline{\mathbf{x_2}}) \qquad \mathbf{c_2} = (\overline{\mathbf{x_3}} \wedge \overline{\mathbf{x_4}})$

- Menge an Schnitten stellt Parallelanordnung dar

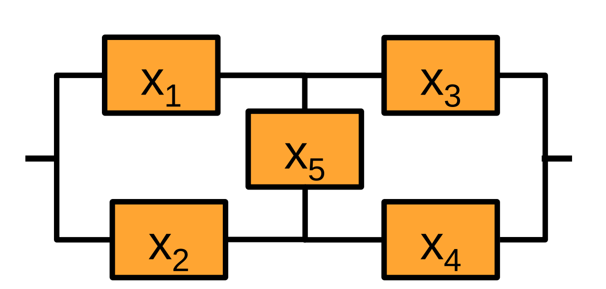
$\mathbf{c_{12}} = (\overline{\mathbf{x_1}} \wedge \overline{\mathbf{x_2}}) \vee (\overline{\mathbf{x_3}} \wedge \overline{\mathbf{x_4}})$



Minimale Ausfallschnitte (2/2)



– Beispiel:



- Schnitt 1: $\mathbf{c_1 = (\overline{x_1} \wedge \overline{x_2})}$
- Schnitt 2: $\mathbf{c_2 = (\overline{x_3} \wedge \overline{x_4})}$
- Schnitt 3: $\mathbf{c_3 = (\overline{x_1} \wedge \overline{x_5} \wedge \overline{x_4})}$
- Schnitt 4: $\mathbf{c_4 = (\overline{x_2} \wedge \overline{x_5} \wedge \overline{x_3})}$

– Boolesche Verknüpfung der Pfade:

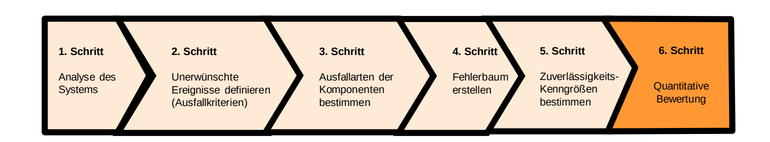
$$\mathbf{f_{ges} = (\overline{x_1} \wedge \overline{x_2}) \vee (\overline{x_3} \wedge \overline{x_4}) \vee (\overline{x_1} \wedge \overline{x_5} \wedge \overline{x_4}) \vee (\overline{x_2} \wedge \overline{x_5} \wedge \overline{x_3})}$$

– Berechnung der Systemausfallwahrscheinlichkeit (Boolesche Modellbildung):

$$\mathbf{F_{ges} = [\big(1-(1-F_1)(1-F_2)\big) * \big(1-(1-F_3)(1-F_4)\big) * \ \dots}$$

$$\mathbf{\dots * \big(1-(1-F_1)(1-F_4)(1-F_5)\big) * \big(1-(1-F_2)(1-F_3)(1-F_5)\big)]}$$

Ergebnisanalyse (1/2)

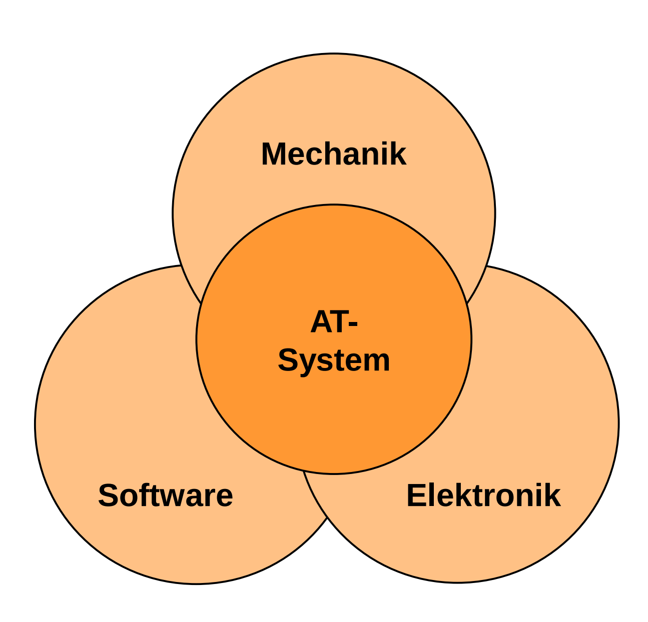


– Nach Berechnung der Zuverlässigkeit oder Systemausfallwahrscheinlichkeit eines unerwünschten Ereignisses erfolgt die Beurteilung hinsichtlich:

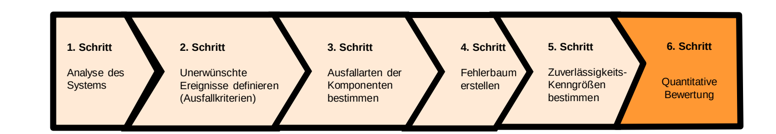
- Mechanik
- Elektronik
- Software bzw. Informationstechnik



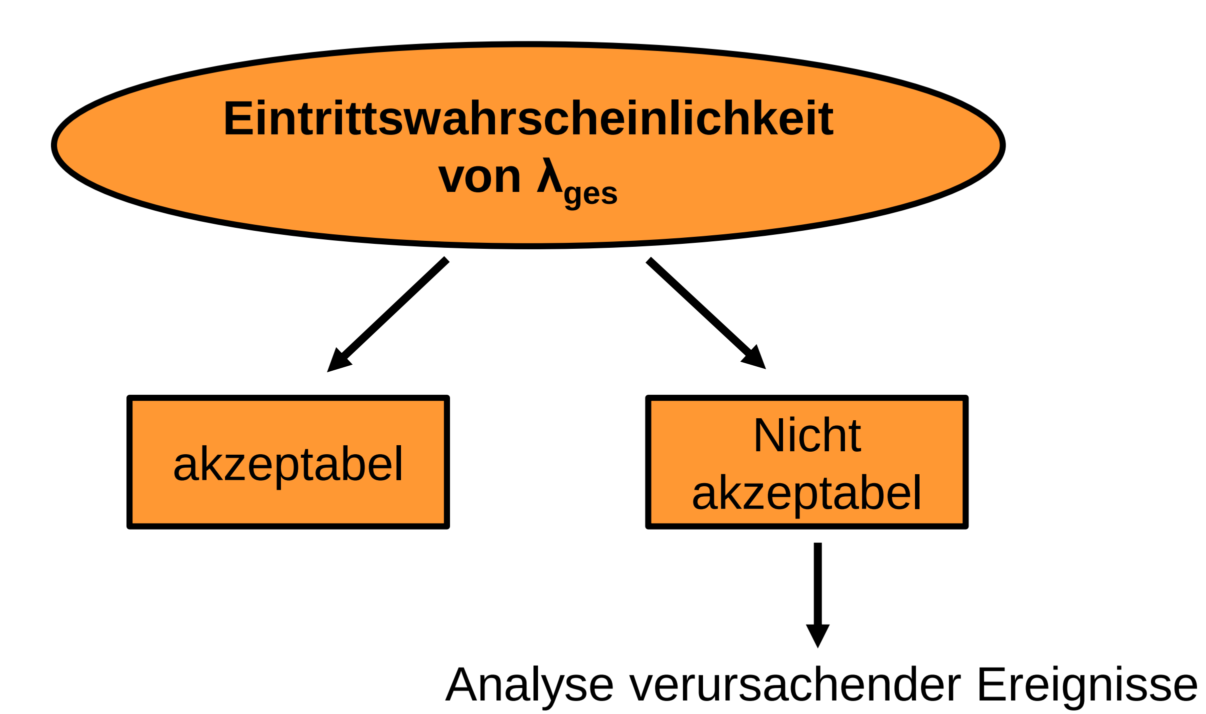
$$\lambda_{ges} = \lambda_{Mech.} + \lambda_{El.} + \lambda_{SW.}$$



Ergebnisanalyse (2/2)



- Durchführung einer Sensitivitätsanalyse als Basis der Ergebnis-Beurteilung durch Entscheidungsträger (Vorgesetzte bzw. Auftraggeber)
- Methode, wie empfindlich Kennzahlen auf eine Änderung der Eingangsparameter reagieren:



Beurteilung der quantitativen Durchführung einer FTA

- Vorteile:
  - Exakte Ergebnisse
  - Große Objektivität und Vergleichbarkeit
  - Relativ geringer Zeit- und Kostenaufwand

- Nachteile:
  - Keine direkt ableitbaren Verbesserungsvorschläge
  - Keine Flexibilität bei der Untersuchung
  - Nicht alle Ausfallursachen erfassbar

Fragen zu Kapitel 3.3

Welchen Aussagen stimmen Sie zu?

☐ Eine quantitative Analyse zeigt stark zuverlässigkeitsbeeinflussende Teile auf.

☐ Quantitativ lassen sich nicht alle Ausfallursachen berücksichtigen.

☐ Funktionsbäume lassen sich mathematisch in Fehlerbäume überführen.

☐ Die Ergebnisse einer quantitativen Bewertung sind eindeutig.

§ 3 Fehlerbaumanalyse (FTA)

Software-FTA

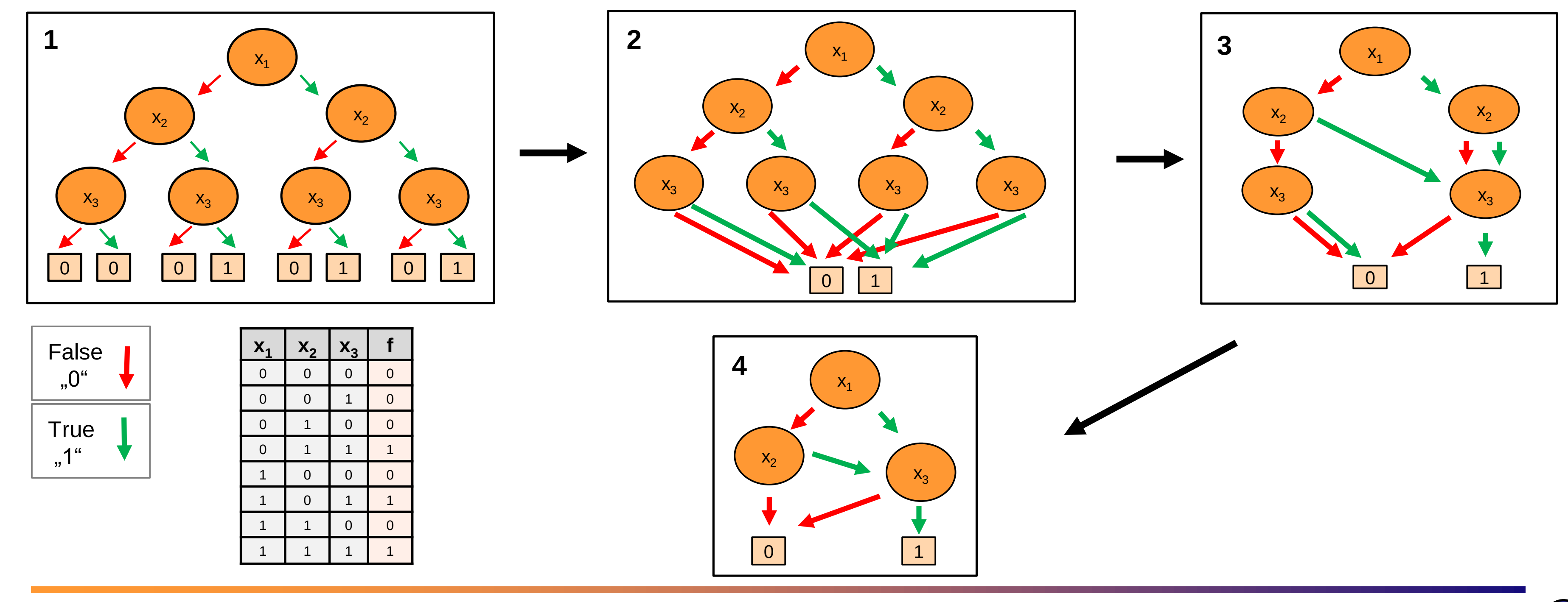
- Anfangs Einsatz der FTA rein zur Bewertung von Anlagen, Schaltungen etc.
- Mit zunehmender Zahl informationstechnischer Systeme ging eine zunehmende Verflechtung der FTA mit Software einher:
    - Software-Tools
        - Unterstützung der Auswertung von Fehlerbäumen
        - Beherrschung komplexer Software-Systeme
    - Entwurf von Software
        - Minimierung des Ausfallrisikos durch Identifikation von fehleranfälligen Modulen (high-risk modules)
        - Kontrolle von Code-Segmenten

Unterstützung bei der Auswertung

- Komplexe Fehlerbäume lassen sich nur unter Anwendung rechnergestützter Methoden sinnvoll behandeln:
  - Qualitativen Bewertung
    - Identifikation kritischer Pfade
    - Erkennen kritischer Mengen
  - Quantitativen Bewertung
    - Aufstellen von Pfaden und Schnitten
    - Berechnung von Zuverlässigkeitskenngrößen
- Lange Rechenzeiten und große Speicherkapazitäten notwendig
- Sorgfältige Planung des Einsatz:
  - Strukturierte Analyse des Systems
  - Effiziente Aufstellung des Fehlerbaums

Beherrschung komplexer Systeme

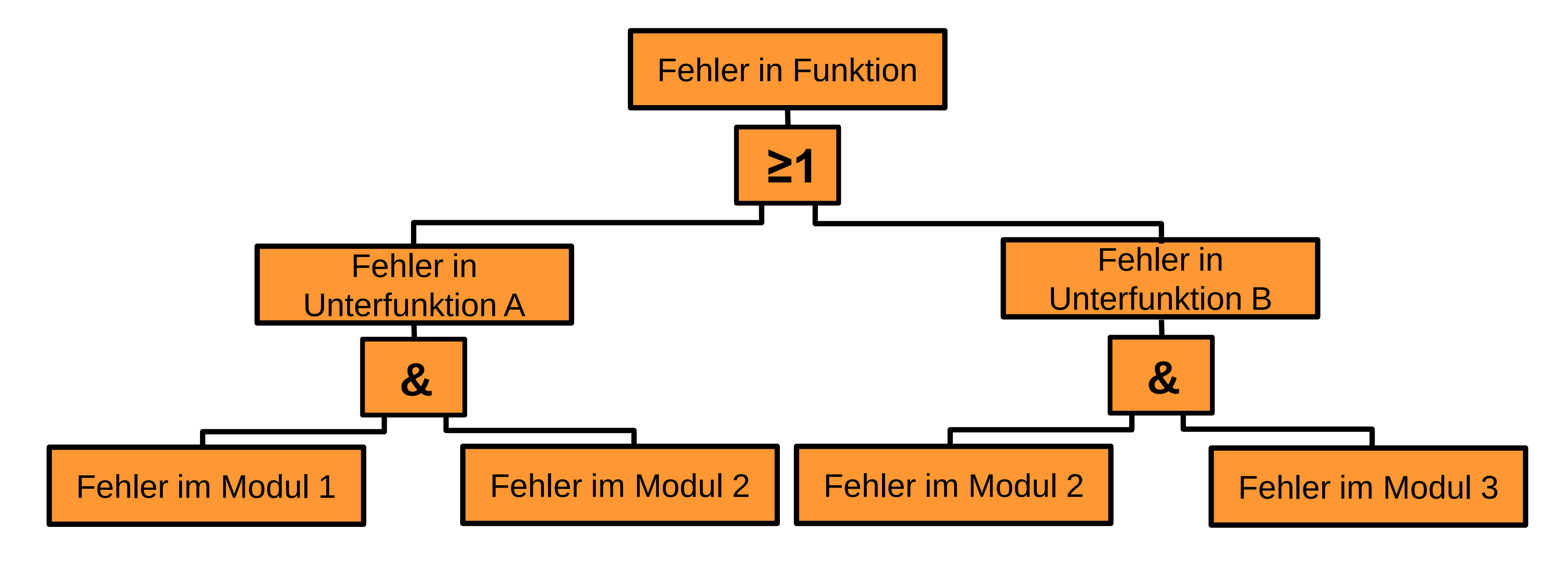
- Fehlerbäume sind häufig unübersichtlich und enthalten Redundanzen
- Auf Basis der Primär-Ereignisse und deren Kombination können Fehlerbäume mathematisch als boolesche Funktionen beschrieben werden, die über Algorithmen vereinfacht werden können (z.B. Verfahren nach Quine und McCluskey)
- Beispiel der Systematik in Form binärer Entscheidungsdiagramme:



x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Minimierung des Ausfallrisikos

– Minimierung des Ausfallrisikos von Software-Systemen erfolgt durch die Identifikation von fehleranfälligen Modulen, sogenannten *high-risk modules*:

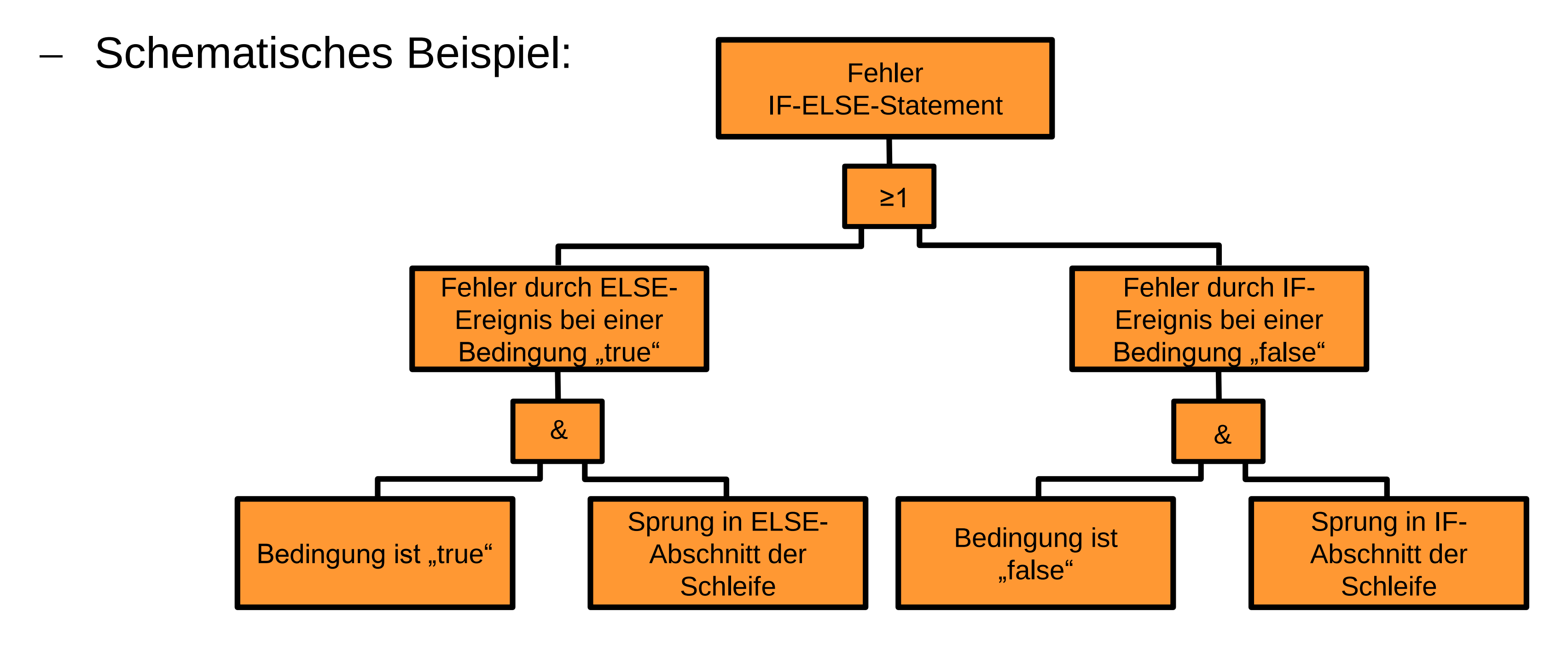


– Ableitung von Maßnahmen zur Absicherung der erkannten Module

– Im Anschluss ausführliche Tests und Re-Design des Systems (falls nötig)

Kontrolle von Code-Segmenten

- Kontrolle einer kompletten Software kaum möglich, da Abhängigkeiten im Code zu komplex sind um diese über einen Fehlerbaum darzustellen
- Daher ist nur die automatische oder manuelle Überführung von Modulen, Funktionen oder Schleifen in Fehlerbäume sinnvoll
- Top-Ereignis wird durch einen unerwünschten Output dargestellt
- Schematisches Beispiel:



Probleme bei der Software FTA

- Fehlerbaum gibt ein System nur zu einem festen Zeitpunkt wieder
  - Aber die Reihenfolge und die zeitlichen Abstände des Auftretens von Ereignissen ist entscheidend über einen Ausfall von Software!
  - Abhilfe über mehrere Fehlerbäume, die einen dynamischen Ablauf darstellen können

- Software-Systeme können verschiedene Zustände einnehmen, wobei erst der weitere Verlauf oder der Kontext angibt, ob dieser Zustand zu einem Ausfall führen kann
  - Aber die FTA erlaubt nur die Zustände „funktionsfähig“ oder „defekt“!
  - Abhilfe bei der quantitativen Durchführung über Wahrscheinlichkeitstheorie auf Basis exakter statistischer Daten

Aussichten

- System werden immer komplexer, sodass eine Auswertung „von Hand“ kaum mehr möglich ist
    - Rechnergestützte Methoden zur Behandlung von Fehlerbäumen
    - Effektive Methoden notwendig, die analytischen und simulativen Charakter aufweisen
    - Kombination der FTA mit anderen Methoden
- Aussagen über Software auf Basis empirischer Daten kaum möglich
    - Genauigkeit und Belastbarkeit der Ergebnisse können in jeder Entwicklungsphase in Frage gestellt werden
    - Erfahrungswerte müssen geschaffen werden
    - Berechnungsverfahren müssen Erfahrungswerte berücksichtigen

Fragen zu Kapitel 3.4

Welchen Aussagen stimmen Sie zu?

☐ FTA ermöglicht die Identifikation von fehleranfälligen SW-Modulen.

☐ Es ist sinnvoll, die FTA mit weitere Analysemethoden zu kombinieren.

☐ Jede Software lässt sich in einen Fehlerbaum überführen.

☐ Ein Fehlerbaum kann immer die Zustände einer Software wiederspiegeln.

Gliederung

§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

Einführung in die FMEA

- Fehlermöglichkeits- und Einflussanalyse, kurz FMEA (**F**ailure **M**ode and **E**ffects **A**nalysis), ist die bekannteste Methode der Zuverlässigkeitsanalyse
- Qualitatives, induktives Verfahren zur Analyse von Ausfällen
- Es erfolgt die Ausfall- und Fehler-Betrachtung nach:
  - Folge
  - Art
  - Ursache
- FMEA ist eine vorbeugende Maßnahme:
  - zur Fehlerprävention
  - zur Fehlerdetektion

Historie (1940er – 1960er)

- 1949 Beschreibung als militärische Anweisung „*MIL-P-1629 – Procedures for Performing a Failure Mode, Effects and Criticality Analysis*"
- 1960er
  - Planung von Kernkraftwerken
  - Einsatz in der Luft- und Raumfahrt (z.B. Apollo Mission der NASA)



Quelle: en.wikipedia.org, 2014

Historie (1970er – heute)



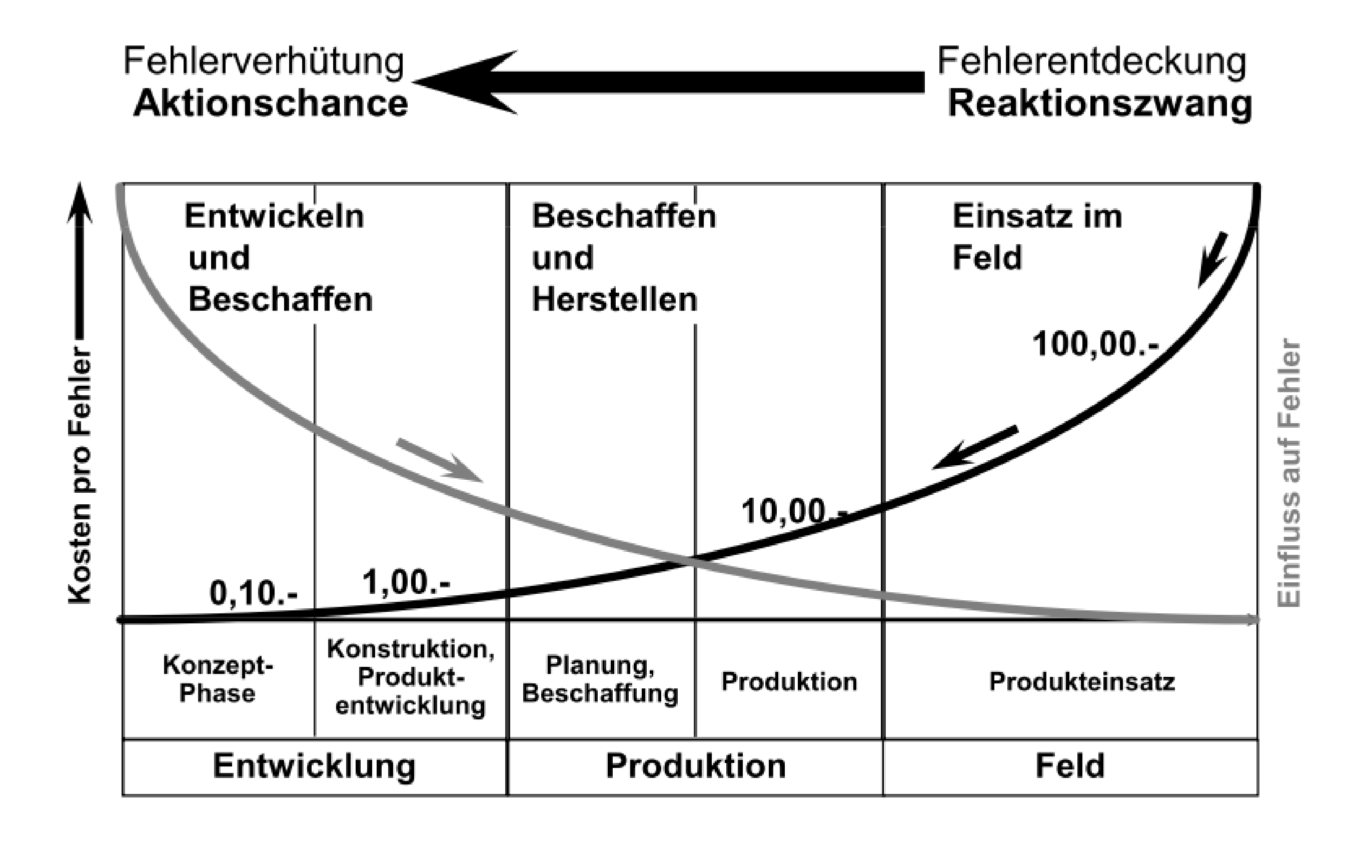
Quelle: www.ceptara.com, 2014

- 1970er – 1980er
  - Einsatz in der Automobilindustrie (Ford in USA)
  - Normung nach DIN 25448 (1980)

- 1990er – heute
  - Einsatz in der Elektronik und Softwareentwicklung
  - Stetige Weiterentwicklungen (System-FMEA, Software-FMEA,…)
  - Normung nach DIN EN 60812 (2006)

Fehlerprävention

– Erkennen und Verhindern von Fehlern in frühen Stadien eines Produktzyklus



Fehlerdetektion

– Detektion möglicher Fehlerquellen, die zu Ausfällen führen können

– Alle Ursachen und resultierenden Folgen von Fehlern auf ein Produkt bzw. System zu ermitteln, zu reduzieren oder gar zu vermeiden

– Fehlerfreie Gestaltung von Prozessen während der Entwicklung

– Schwachstellen von System, Produkten oder Prozesse identifizieren, sodass eine konstruktive Überarbeitung erfolgen kann

Zeitpunkt des Einsatz

- „So früh wie möglich“:
  - Direkt bei Lasten- oder Pflichtenhefterstellung
  - In frühen Phasen des Produktentstehungsprozess (nach ersten Entwürfen)

- „Entwicklungsbegleitend“:
  - Mit permanenter Anpassung und Aktualisierung
  - Als dynamisches Dokument

Einflussfaktoren auf den Einsatz

- Äußere Einflussfaktoren:
  - Gestiegene Kundenanforderungen (an Qualität, Funktionalität, Benutzbarkeit,…)
  - Druck der Kostenoptimierung durch Wettbewerb
  - Gesetzliche Produkthaftung
- Innere Einflussfaktoren:
  - Zeitpunkt der Durchführung
  - Freiheitsgrade im Ablauf
  - Erfahrung der FMEA-Teilnehmer

Ziele der FMEA

- Am System bzw. Produkt:
    - Steigerung der Zuverlässigkeit
    - Einhaltung der Garantiezeiten
    - Funktionssicherheit

- In der Entwicklung und Produktion:
    - Verbesserte Kommunikation zwischen den Beteiligten
    - Effektivere Prozesse
    - Reduzierung von Kosten
    - Reibungslose Serienanläufe in der Fertigung

Vergleich FMEA vs. qualitative FTA

- FMEA und qualitative FTA sind grundlegend unterschiedliche Methoden
- Methoden schließen sich aber gegenseitig nicht aus, sondern ergänzen sich
- Es gilt:

FMEA	Qualitative FTA
Induktive Methode: **Ursache ⇐ Wirkung**	**Deduktive Methode:** **Ursache ⇒ Wirkung**
Kombination von Fehlerursachen und Fehlerfolgen über Systemebenen hinweg.	**Systematische Suche nach Fehlerursachen über Ereignis-Verknüpfungen innerhalb von Systemebenen**

Frage zu Kapitel 4.1

Welchen Aussagen stimmen Sie zu?

☐ Der Durchführungszeitpunkt einer FMEA beeinflusst das Ergebnis.

☐ Ein Ziel der FMEA ist die fehlerfreie Gestaltung von Prozessen.

☐ Die FMEA ist eine induktive Zuverlässigkeitsmethode.

☐ FMEA entspricht prinzipiell der qualitativen FTA.

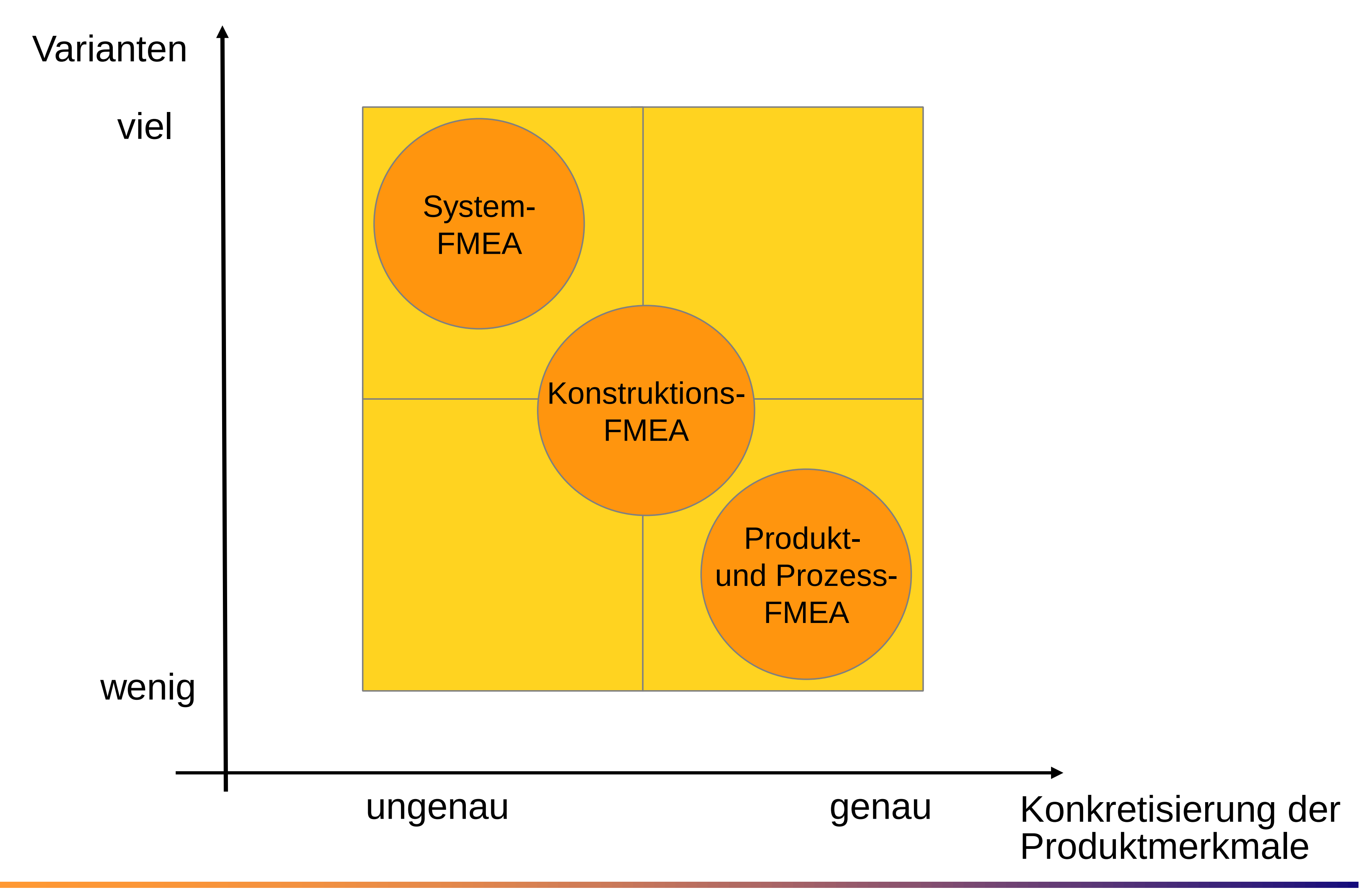
§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

Unterscheidung nach Arten

- Im Entwicklungsprozess wird nach drei FMEA-Arten unterschieden:
    - System-FMEA
    - Konstruktions-FMEA (auch als Design-FMEA bezeichnet)
    - Produkt- und Prozess-FMEA

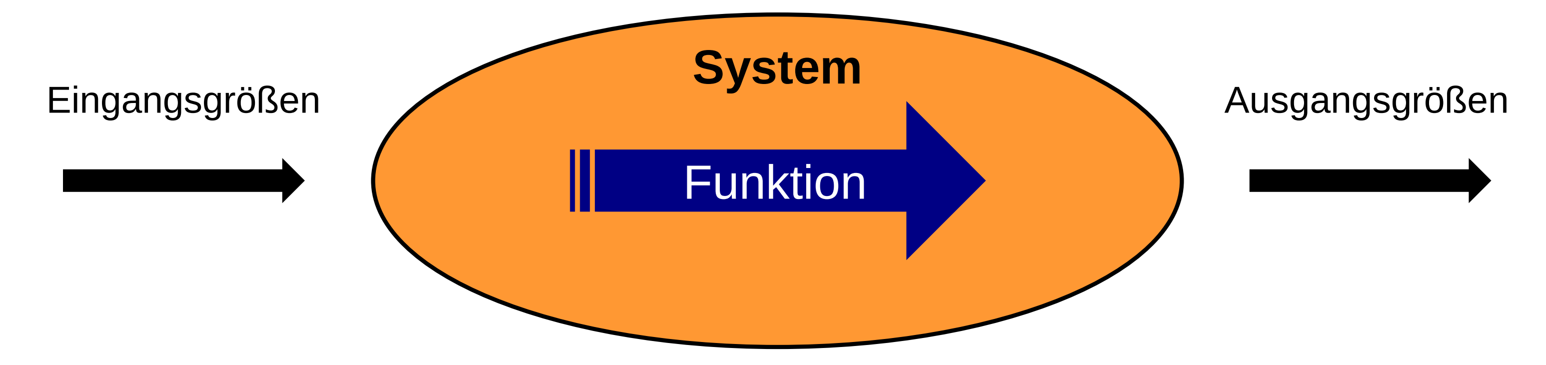
- Aufteilung in Arten notwendig, da:
    - Oft funktionale Zusammenhänge zwischen Komponenten nicht korrekt erfasst werden können
    - Aufgrund steigender Komplexität keine Analyse des gesamten Herstellungsprozess von der Planung bis hin zur Auslegung von Systemen mehr möglich ist

FMEA-Arten im Entwicklungsfeld



Begriffe System und Funktion

- „System“ ist ein technisches Gebilde (Maschine, Gerät, Anlage,…):
  - Klar abgrenzbar von der Umgebung
  - Untergliederung in Teilsysteme möglich
  - Einteilung in Einheiten möglich (Baugruppen, Komponenten,…)
- „Funktion“ beschriebt den Zusammenhang zwischen dem Ein- und Ausgangsgrößen von Systemen:
  - Aufgabenbeschreibung
  - Eindeutige Zuordnung möglich



System-FMEA

- Ziel ist es, mögliche funktionale Fehler in einem frühen Stadium zu identifizieren:
    - Strukturierung des zu untersuchenden Systems in Einheiten
    - Aufzeigen von funktionalen Zusammenhängen zwischen Einheiten
    - Ableitung von Fehlfunktionen der Einheiten
    - Logische Verknüpfung der Fehlfunktionen
    - Abgleich des Ergebnis mit dem Pflichtenheft

- Bildet die Grundlage der Konstruktions-FMEA
- Wird von der Entwicklungsabteilung durchgeführt

Konstruktions-FMEA

- Ziel ist der einwandfreie Entwurf eines Produkts bzw. Systems unter Betrachtung der einzelnen Komponenten:
  - Fertigungseignung in frühen Phase abschätzen
  - Identifikation systematischer Fehler in der Konstruktion
  - Vermeidung von Entwicklungsfehler
- Spezielle Unterteilung möglich in:
  - Hardware-FMEA zur Analyse von Hardware und Elektronik
  - Software-FMEA zur Analyse von Programmcode
- Bildet die Grundlage der Produkt- und Prozess-FMEA
- Wird von der Konstruktionsabteilung durchgeführt

Produkt- und Prozess-FMEA

- Ziel ist die Elemination von identifizierten Fehlern am Produkt selbst und am produzierenden Prozess
- Am Produkt:
  - Betrachtet werden physikalische Ausfallarten (Bruch, Verschleiß,…)
  - Betrachtung erfolgt über alle Systemhierarchien hinweg
- Am Prozess:
  - Betrachtet werden Fehler im Produktionsprozess (Fertigung, Montage, Logistik, Transport,…)
  - Strukturierte Beschreibung nach den „5 M‘s“ (**M**ensch, **M**aschine, **M**aterial, **M**ethode, **M**ilieu/**M**itwelt)

- Wird von der Planung und Produktion durchgeführt

Übersicht der FMEA-Arten

Art	Betrachtungs-gegenstand	Eingangsgröße	Durchführender Bereich/Abteilung
System-FMEA	Systemebene bzw. übergeordnetes Produkt	Pflichtenheft und Produktkonzept	Entwicklung
Konstruktions-FMEA	Bauteile bzw. Komponenten	Konstruktions-unterlagen und Ergebnis der System-FMEA	Konstruktion
Produkt- und Prozess-FMEA	Produkt und Fertigungsschritte	Fertigungs- bzw. Montageabläufe und Ergebnis der Konstruktions-FMEA	Planung und Produktion

Frage zu Kapitel 4.2

Welchen Aussagen stimmen Sie zu?

☐ Eine System-FMEA ist die Grundlage für eine Konstruktions-FMEA.

☐ Die System-FMEA schätzt die Fertigungseignung eines Produkts ab.

☐ Die Produktion führt die Konstruktions-FMEA durch.

☐ Produkt-FMEA befasst sich mit physikalischen Fehlern.

§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

Werkzeuge bei der Durchführung einer FMEA

- Bei der FMEA wird ein Betrachtungsgegenstand so lange unterteilt, bis er als geschlossene Funktionseinheit betrachtet werden kann
- Sehr komplex bei größeren Systemen
- Einsatz von Werkzeugen notwendig, um Komplexität zu beherrschen
- Beispiele :
  - Fehlerbaumanalyse (FTA)
  - Fischgräten- bzw. Ursache-Wirkungs-Diagramm
  - Fragenkatalog bzw. Checklisten
  - Ereignisablaufanalyse (ETA)
  - Matrix-Diagramm
  - FMEA-Software

Fehlerbaumanalyse (FTA)

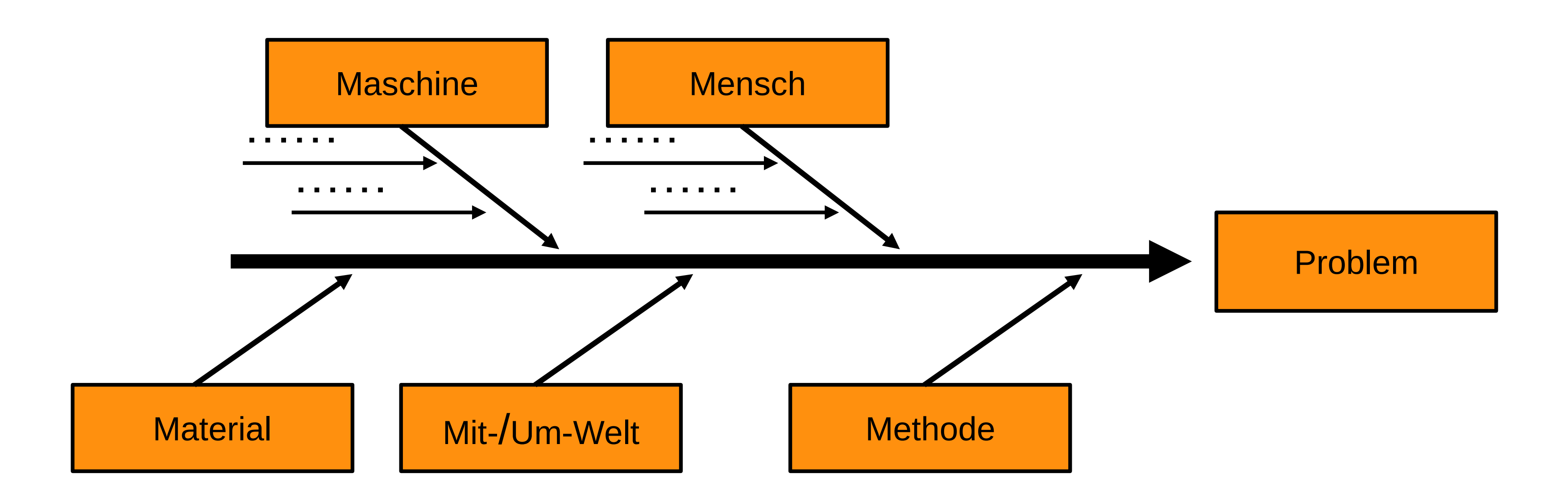
- Grundlagen und Durchführung siehe Kap.3

- Anwendung erfolgt in Form der qualitativen FTA

- Ziel der FTA im Rahmen der FMEA ist es, nicht nur Ausfallarten bzw. Ausfallursachen eines Systems zu detektieren, sondern speziell:
  - Funktionale Zusammenhänge über Verknüpfungen herzustellen
  - Auswirkungen auf das System beschreiben
  - Systematische Strukturierung herzustellen

Fischgräten- bzw. Ursache-Wirkungs-Diagramm

– Darstellung der Kausalität zwischen einem unerwünschten Ereignis (Wirkung) und den Gründen, die zu diesem Ereignis führen (Ursachen)

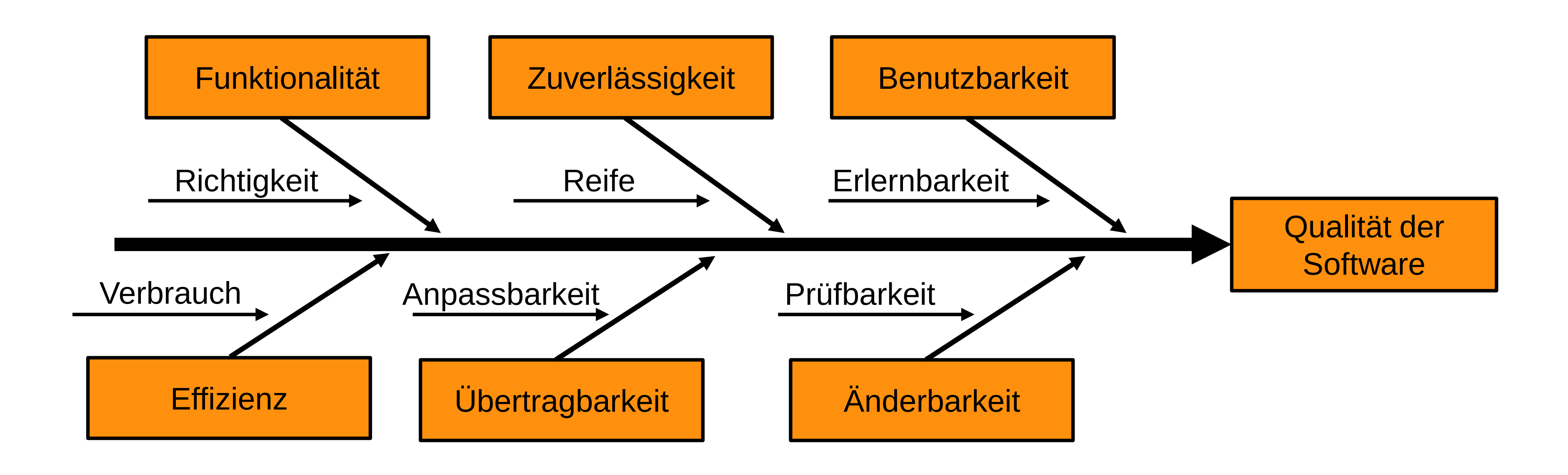


Ursache-Wirkungs-Diagramm - Einsatz und Bewertung

- Einsatzmöglichkeiten:
  - Analyse von Fehler- und Ausfallursachen
  - Prozesse zur Produktivitätssteigerung
  - Problemen innerhalb bzw. zwischen Abteilungen
  - Optimierung eines Systems
- Vorteile:
  - Übersichtliche und interdisziplinare Darstellung
  - Einfache Erlern- und Lehrbarkeit (geringer Zeit- und Kostenaufwand)
- Nachteile:
  - Zu umfangreich bei komplexen Problemstellungen
  - Wechselwirkungen oder Zeitabhängigkeiten nicht erfassbar

Ursache-Wirkungs-Diagramm - Anpassung an Software

- Adaption und Einsatz des Ursachen-Wirkungs-Diagramms für die FMEA von Software-Systemen möglich
- Beispiel:



(Prüf-)Fragenkatalog bzw. Checklisten

- Zusammenstellung von Fragen zur Analyse von Systemen, Prozessen und Arbeitsmitteln
- Ermöglicht eine systematische Bewertung des Ist-Zustands
- Beispielhafter Auszug zu einem unerwünschten Ereignis „µC liefert keine Daten“:

Ebene	Kriterium	Ergebnis	Anmerkung
Hardware	Spannungsversorgung angeschlossen?	Ja/Nein	-
	Pins korrekt beschaltet?	Ja/Nein	Abgleich mit Datenblatt
	Lötstellen in Ordnung?	Ja/Nein	Prüfung: Bildverarbeitung
Software	Schleifen korrekt verschachtelt?	Ja/Nein	Kontrollflussgraph
	Register korrekt initialisiert?	Ja/Nein	Abgleich mit Datenblatt
	Flags richtig gelöscht?	Ja/Nein	-
Schnittstelle	Abschlusswiderstände an Leitungen?	Ja/Nein	z.B. 120 Ohm bei CAN

Prüffragenkatalog - Einsatz und Bewertung

- Bestehende Fragenkataloge können von Gewerkschaften, Berufsverbänden oder Industrieverbänden angefordert werden
- Einsatz vor der FMEA, da Aussagen über Komponenten ableitbar, die bei der Durchführung genauer untersucht werden müssen
- Vorteil:
  - Zielgerichtetes Vorgehen durch klare und nachvollziehbare Abläufe
  - Hinweise auf realtypische, auch komplexe Mängel und Fehlerarten
  - Arbeitshilfe zur Dokumentation und Qualitätssicherung
- Nachteil:
  - Qualität hängt von der Erfahrung der Ersteller ab
  - Keine Angaben über neu entstandene, nicht berücksichtigte oder noch unbekannt Fehlerursachen möglich

Ausschnitt Prüffragenkatalog „Bildschirmarbeitsplätze"

Prüfliste "Bildschirmarbeitsplätze" P 9

Prüfliste Bildschirmarbeitsplätze

Werden Fragen mit "Nein" beantwortet, sind geeignete Maßnahmen zu veranlassen, um Unfall- und Gesundheitsgefahren an diesen Arbeitsplätzen vorzubeugen.

Arbeitsplatz/Arbeitsumgebung

1. Hat der Arbeitstisch, je nach Aufgabe (zusätzliche Arbeitsmittel, wechselnde Tätigkeiten) eine ausreichende Arbeitsfläche? (Breite: 120 cm – 160 cm; Tiefe: mind. 80 cm)
2. Ist der Arbeitstisch mindestens so tief, dass vor der Tastatur ein Freiraum von 5 bis 10 cm Tiefe zur Auflage der Handballen verbleibt und gleichzeitig der Sehabstand zum Bildschirm 60 cm nicht überschreitet und das Bildschirmgerät nicht über die hintere Kante der Tischplatte hinausragt?
3. Ist der Arbeitstisch höhenverstellbar? Falls nein: Sind nicht höhenverstellbare Tische 72 cm hoch?
4. Ist die Tischoberfläche reflexionsarm?
5. Ist ein Vorlagenhalter vorhanden, verschiebbar und in Höhe und Neigung verstellbar?
6. Sind die notwendigen Fußstützen vorhanden (z.B. bei nicht höhenverstellbaren Tischen für Beschäftigte mit unterdurchschnittlicher Körpergröße)?
7. Steht ein höhenverstellbarer Bürodrehstuhl mit verstellbarer Rückenlehne zur Verfügung?
8. Ist eine ausreichend blend- und flimmerfreie Beleuchtung von mindestens 500 Lux vorhanden?
9. Sind Langfeldleuchten parallel zu den Fenstern angebracht und sorgen für eine gleichmäßige Helligkeit?
10. Ist die vorwiegende Blickrichtung parallel zu den Fenstern?
11. Ist der Bildschirm frei von Spiegelungen, Blendungen, Reflexionen?
12. Lassen sich die Fenster abdunkeln, z.B. durch Jalousien oder Vorhänge?

Quelle: www.bghw.de, 2014

Bildschirmgerät und Tastatur

24. Sind Bildschirmgerät und Tastatur mit dem CE- oder GS-Zeichen gekennzeichnet? Falls "Nein", so ist beim Hersteller eine schriftliche Bestätigung einzuholen, dass Gerät und Tastatur den zum Zeitpunkt der erstmaligen Inbetriebnahme geltenden Arbeitsschutzbestimmungen entsprechen.
25. Ist die Tastatur variabel aufstellbar und die Beschriftung gut lesbar?

Unterweisung

26. Sind die Beschäftigten über die richtige Einstellung der Arbeitsmittel, die Reinigung der Bildschirmgeräte und Tastatur sowie über die richtige Sitzhaltung unterrichtet?
27. Ist die Unterweisung dokumentiert?
28. Wird die Unterweisung unter Berücksichtigung des festgestellten Fehlverhaltens sowie bei körperlichen Beschwerden, die auf die Tätigkeit am Bildschirmarbeitsplatz zurückgeführt werden können, wiederholt?
29. Werden die Beschäftigten - insbesondere bei der Nutzung eines Arbeitsplatzes durch verschiedene Personen - angehalten, die Arbeitsmittel richtig einzustellen und zu benutzen?

Software

30. Enthält die Software alle für die Aufgaben der Benutzer benötigten Funktionen?
31. Können die Benutzer ohne Umwege und Tricks ihre Arbeitsergebnisse erzielen?
32. Sind die Informationen, die für die Benutzer zur Erledigung ihrer Aufgaben notwendig sind, auf dem Bildschirm übersichtlich verfügbar?
33. Sind die Meldungen des Systems für die Benutzer immer verständlich?

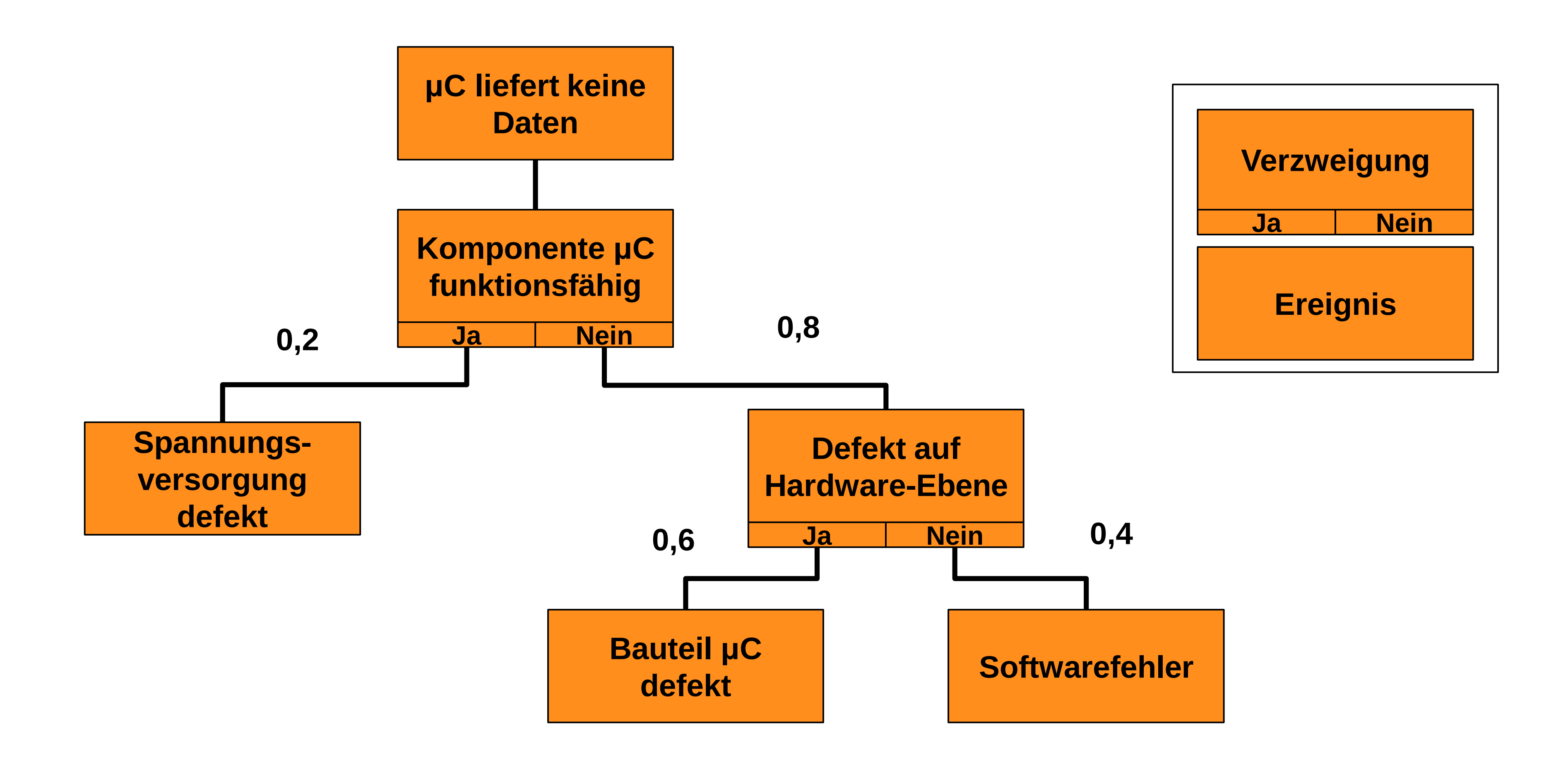
Ereignisablaufanalyse (ETA)

- Ereignisablaufanalyse (ETA, **E**vent **T**ree **A**nalysis) ist ein induktives Verfahren, bei dem mögliche Folgen eines Fehlers bestimmt werden sollen
- Normung nach DIN 25419
- Vorgänge bzw. logische Zusammenhänge werden graphisch in verzweigten Ereignisbäumen dargestellt (über Ja/Nein-Bedingung)
- Ereignisablaufanalyse gleicht auf ersten Blick der Fehlerbaumanalyse, aber:
  - FTA: Ermittlung aller Ursachen, die zu einer bestimmten Gefährdung führen können (deduktiv)
  - ETA: Ermittlung aller Gefährdungen, die auf einen bestimmten Fehler bzw. Störfall folgen können (induktiv)

ETA - Einsatz und Bewertung

- Sinnvoller Einsatz frühestens dann wenn alle Systemanforderungen bekannt, sodass alle Verzweigungen berücksichtigt werden können
- Praxis häufig in Kombination mit FTA im Rahmen der probabilistischen Sicherheitsanalyse PSA (vgl. Kap 6)
- Vorteil:
  - Ermöglicht vollständige Sicherheitsbetrachtung nach einem einfachem Schema (Ja oder Nein)
  - Zuordnung von Wahrscheinlichkeiten zur quantitativen Auswertung möglich
- Nachteil:
  - Unübersichtliche Bäume bei komplexen Problemstellungen
  - Erstellung und Auswertung daher oft nur über Computerprogramme möglich

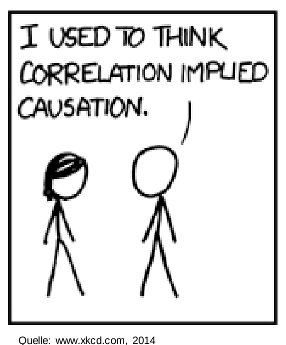
Beispiel einer ETA zu „µC liefert keine Daten"

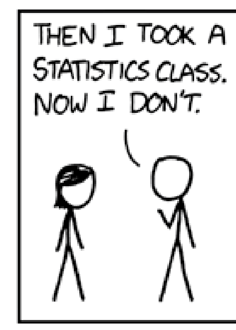


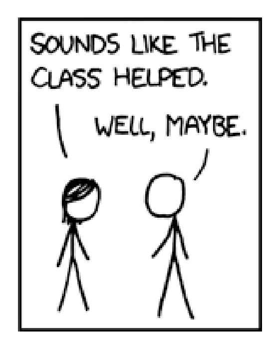
Matrix-Diagramme – Korrelation und Kausalität

– Graphische Darstellung von Korrelationen zwischen Betrachtungs-Gegenständen und/oder ihren spezifischen Eigenschaften

– Häufiger Irrtum:

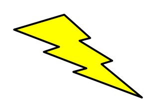






Quelle: www.xkcd.com, 2014

- Korrelation ist eine Wechselbeziehung, d.h., dass sich beispielsweise Merkmale, Funktionen oder Ereignisse ähnlich verhalten:
  „Im Sommer wird mehr Eis gegessen und es gibt mehr Sonnenbrände."

- Kausalität ist das Prinzip zwischen Ursache und Wirkung, d.h., es gibt eine Richtung von Aktion zu Reaktion:
   *„Eis essen verursacht Sonnenbrand."*

Matrix-Diagramm – Einsatz und Bewertung

- Bei der FMEA erfolgt die Verbindung von Komponenten zu den einzelnen, produktsicherheitsrelevanten Merkmalen und Sichtweisen auf das Produkt (Problemursache, Verantwortlichkeiten, Ressourcen,…)
- Es erfolgt Überführung und Gewichtung in Tabellenform:

Merkmale	Komponenten
	Gewichtung

- Vorteil:
  - Klare Übersicht
  - Besseres Verständnis
- Nachteil:
  - Erfordert Erfahrung und Fachverständnis bei großen und komplexen Systemen

Matrix-Diagramm – Vorgehensbeispiel der Tabellenform

- Merkmale in den Zeilen werden mit Faktoren (z.B. 1-3) gewichtet
- Zeilenwerte einer Komponente werden in der Spalte addiert
- Komponente mit dem höchsten Zahlenwerten birgt das größte Ausfallrisiko
- Komponente mit dem größten Risiko wird in der FMEA detaillierter untersucht

Kosten-Kriterien	Bauteil / Systemelement		
	Sensor	Verstärker	µC
Neuentwicklung	2	1	3
Produktänderung	2	1	3
Prozessänderungen	2	2	2
Summe	6	4	8

FMEA-Software

- Nutzen der EDV-Unterstützung:
  - Einfache Erstellung, Pflege und Dokumentation
  - Unterstützung des FMEA-Teams (besonders Moderator, vgl. Kap 4.4)
  - Überwachung der Terminplanung und Verantwortlichkeiten
  - Gute Darstellung der Schritte bzw. Ergebnisse während der Durchführung

- Zusätzlicher Aufwand:
  - Auswahl einer geeigneten Software gemäß den Anforderungen des Unternehmens bzw. des FMEA-Teams
  - Erlernen und Beherrschen des FMEA-Programms

Software Beispiel: IQ-FMEA



Quelle: www.apis.de, 2014

– „IQ-FMEA“ von *APIS Informationstechnologien*

Quelle: www.apis.de, 2014

Quelle: www.apis.de

Software Beispiel: PLATO SCIO FMEA



Quelle: www.plato.de, 2014

– „PLATO SCIO-FMEA“ von *Plato GmbH*

Input

Risikobetrachtung mit PLATO SCIO™-FMEA

Output

SCIO™ Fehlerbaum-analyse

SCIO™ Anforderungen

Normen

Gesetze

Strukturen

SCIO™ Block-Diagram

Risikoanalyse

Risiko-bewertung

Risiko-minimierung

AQTIO™ Maßnahmen-management

SCIO™ Control Plan Prüfplan

SCIO™ Produkt-FMEA Prozess-FMEA

SCIO™ Prozessplanung

XERI™ Dokumentation Produkthaftung

XERI™ Dokumentation Risiko-Managementakte

XERI™ Dokumentation QS 9000, PPAP, VDA

Quelle: www.plato.de, 2014

Frage zu Kapitel 4.3

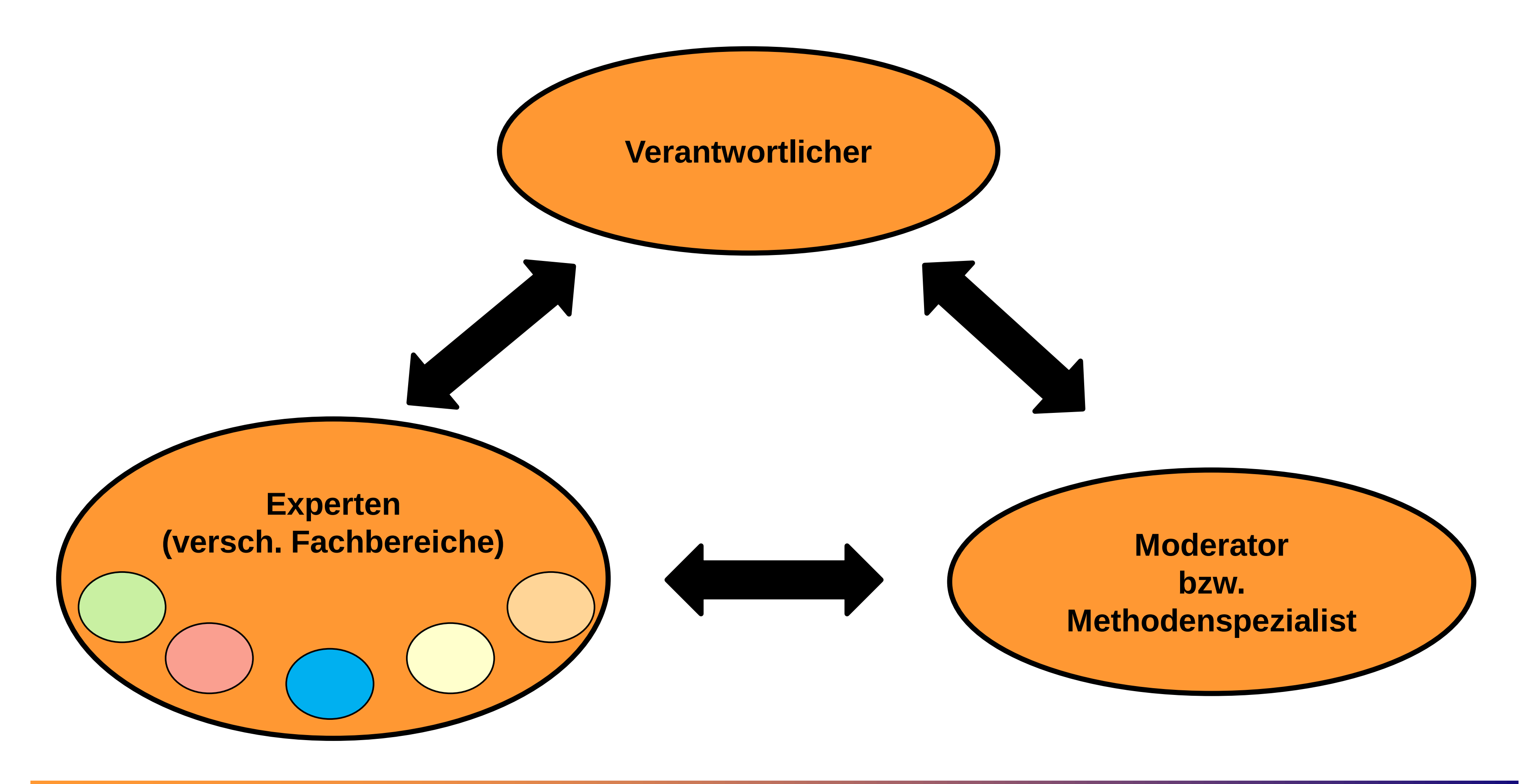
Welchen Aussagen stimmen Sie zu?

- ☐ FTA entspricht der Ermittlung aller Gefährdungen, die auf einen bestimmten Fehler bzw. Störfall folgen können.
- ☐ In Matrix-Diagrammen werden Kausalitäten von Komponenten verglichen und bewertet.
- ☐ Bei Fischgräten-Diagrammen können auch zeitbehaftetet Ausfallarten korrekt dargestellt werden.
- ☐ Checklisten garantieren eine qualitative Zuverlässigkeitsanalyse auch von Entwicklern, die noch über keine große Erfahrung verfügen.

§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

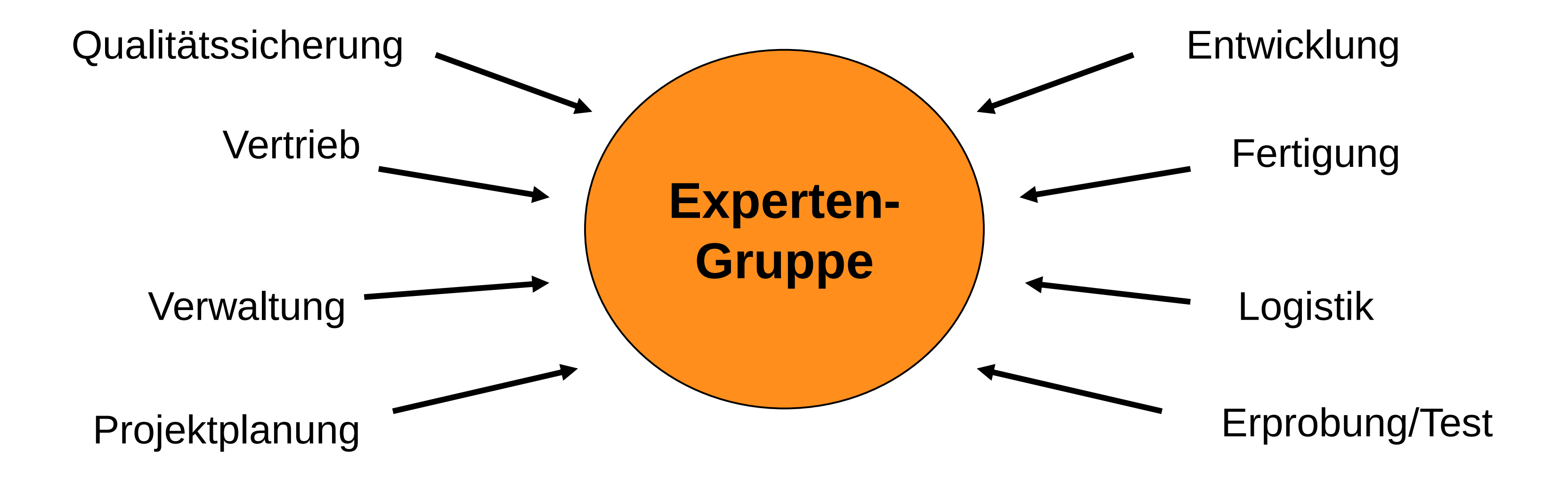
Teamorientierung

- FMEA ist immer teamorientiert
- Einzelnen Punkte werden von einem FMEA-Team diskutiert und erörtert



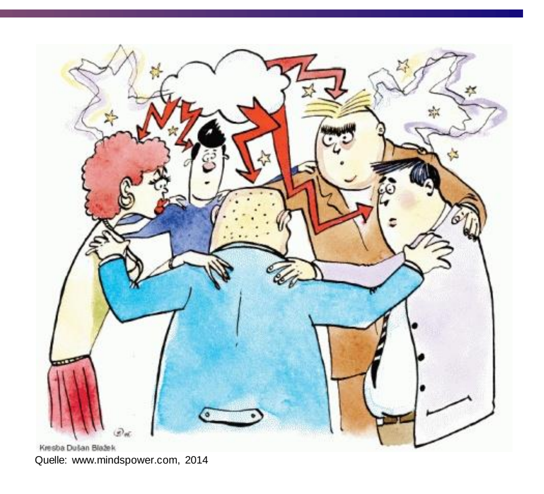
Beteiligte Fachbereiche

– Expertengruppe setzt sich aus Vertretern verschiedener Fachbereiche zusammen:



– Moderator ist ein Methodenspezialist aus einem der Fachbereiche
– Moderator kann auch identisch mit dem Verantwortlichen sein
– Verantwortlicher ist der Initiator der FMEA, oftmals Gesamtprojektleiter

Aufgabe des FMEA-Team (1/2)



Quelle: www.mindspower.com, 2014

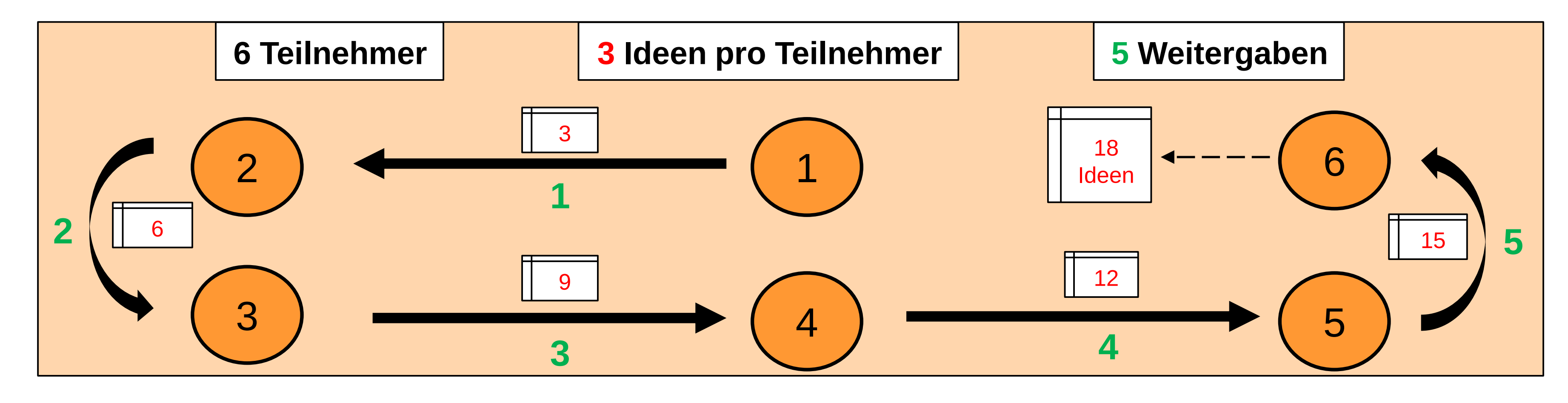
- Problemlösung und Ideenfindung durch gegenseitige Anregung auf möglichst intuitive Art
- Laute Techniken, z.B. Diskussion über *Brainstorming*:
  - Nach Alex F. Osborn, Autor, USA 1939
  - Gruppe von 5 – 7 Personen, je nach Problemstellung Experten/Laien



  - Grundsätzliche Regeln:
    - Freies Aufgreifen von Ideen, freies Assoziieren und Phantasieren
    - Kommentare, Korrekturen und Kritik sind verboten
    - Begrenzter Zeitrahmen von ca. 5 - 30 min
  - Einfache und kostengünstige Methode
  - Gruppendynamik /-synergie führt oftmals zu innovativen Lösungsansätzen
  - Nachteilig ist die Abhängigkeit von den Teilnehmern, Gefahr des Abschweifens und die Selektion geeigneter Ideen

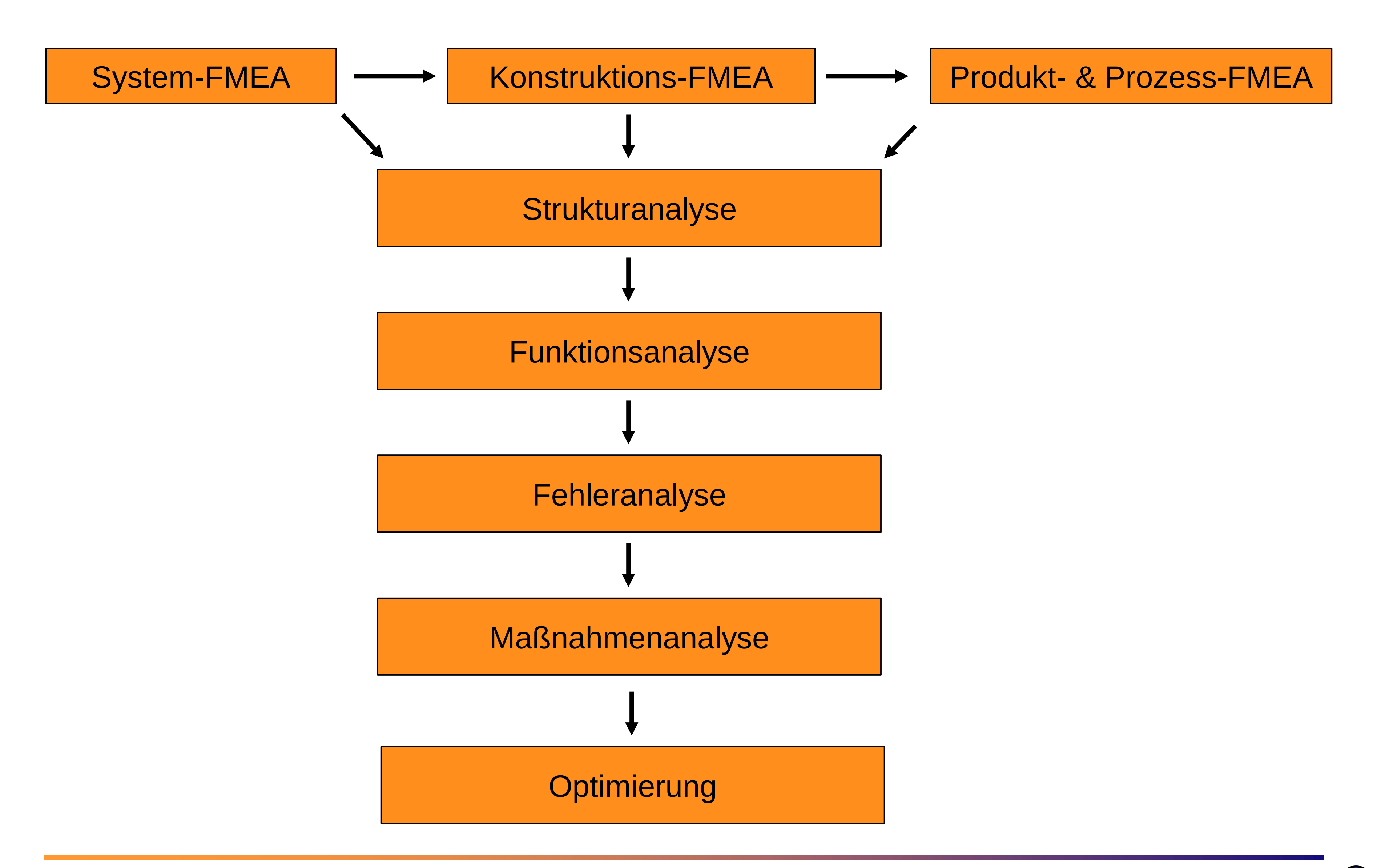
Aufgabe des FMEA-Team (2/2)

– Ruhige Techniken, z.B. Kreativitätstechnik *Brainwriting* über „6-3-5-Methode“:
  - Nach Bernd Rohrbach, Unternehmensberater, Deutschland 1968
  - Reihenweise werden Ideen nach dem Schema niedergeschrieben, dass bereits notierte Ideen aufgegriffen und ergänzt werden:



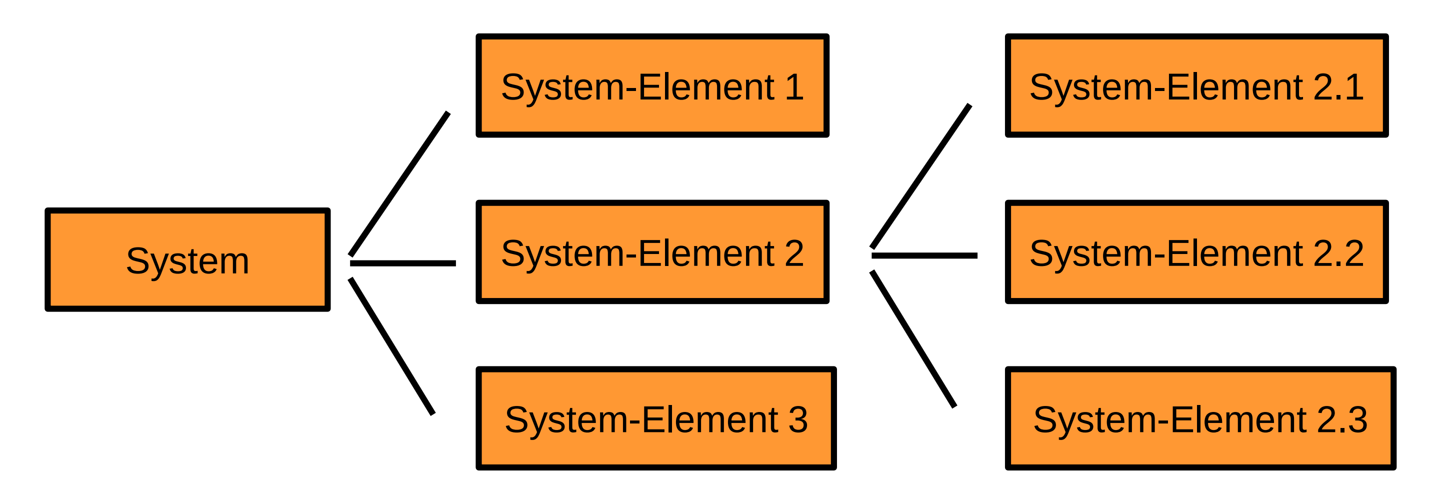
  - Viele Ideen in kurzer Zeit, die nicht „zerredet“ werden
  - Jeder Teilnehmer ist gleichberechtigt
  - Nachteilig ist, dass der starre Ablaufmechanismus die Kreativität stören oder der Bearbeitungstakt für Teilnehmer zu langsam/schnell sein kann

Grundsätzlicher Ablauf einer FMEA



Strukturanalyse

- Systemstruktur erstellen:
  - Zu untersuchendes System wird von der Umgebung abgegrenzt und Schnittstellen werden klar definiert
  - System wird in immer weitere Systemelemente aufgeteilt, bis einzelne Funktionsgruppe, Baugruppen oder Bauteile vorliegen
  - Einzelne Systemelemente werden hierarchisch angeordnet
- Schema:

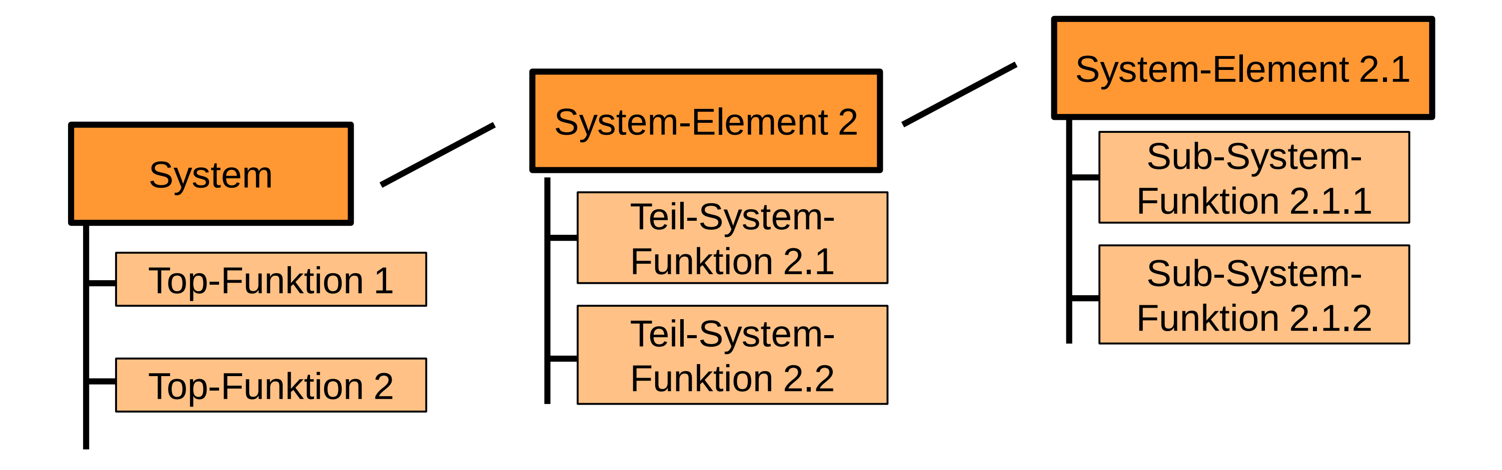


Funktionsanalyse

– Auf Basis der Strukturanalyse werden den Systemelementen einzelne Funktionen zugeordnet

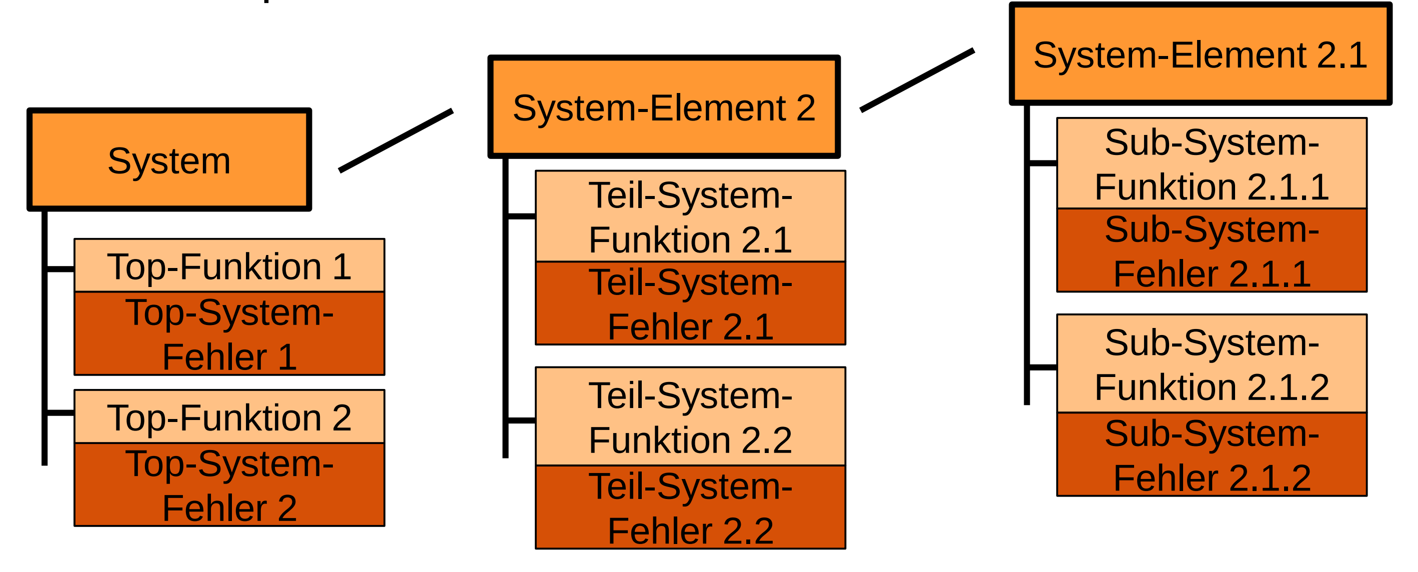
– Vorgehen nach Top-Down-Methode:

- Ausgehend von den Top-Funktionen des Systems werden die entsprechenden Funktionalitäten mit den Elementen verknüpft
- Kenntnis der Funktionsanforderung unverzichtbar (Lasten-/Pflichtenheft)



Fehleranalyse

- Für jedes Systemelement wird eine Fehleranalyse durchgeführt, d.h., mögliche Fehlfunktionen werden ermittelt
- Funktionsfehler können über die Negation der Funktionalität abgeleitet werden
- Entsprechend der Top-Down-Methode werden die Fehlfunktionen mit den Funktionen verknüpft



Maßnahmenanalyse

- Es wird nun der aktuelle Stand des Systems bewertet
- Dazu werden die Kombinationen aus Funktion und Fehler unter Einbeziehung der bisherigen festgelegten Prüfmaßnahmen untersucht:
  - Vermeidungsmaßnahmen zur Risikobeseitigung
  - Entdeckungsmaßnahmen zur Risikominderung
- Das Ergebnis der Untersuchung wird über folgende Kriterien bewertet:
  - Auftretenswahrscheinlichkeit der Fehler-Ursache
  - Bedeutung der Fehler-Folge
  - Entdeckungswahrscheinlichkeit der Fehler-Ursache
- Aus den Bewertungskriterien wird Risikoprioritätszahl berechnet (siehe Kap 4.5)
- Entsprechend der Risikoprioritätszahl erfolgt die Ableitung von weiteren Maßnahmen zur Risikobeseitigung/-minderung

Optimierung

- Optimierungen an den Systemelementen, über die zuvor abgeleiteten zusätzlichen Maßnahmen, erfolgt je nach Priorität:
  - Konzeptänderungen um Fehler-Ursachen auszuschließen
    - Änderung der Konstruktion
    - Umgestaltung von Prozessen
  - Konzeptzuverlässigkeit erhöhen um Schwere der Fehler-Art zu reduzieren
    - Redundanz
    - Fehleranzeigen
  - Wirksamere Entdeckungsmaßnahmen
    - Größere Anzahl von Tests
    - Reviews
- Abschließend wird der neue bzw. geänderte Stand des Systems bewertet
- Abschätzung des Systemzustands erfolgt ernuet über die Risikoprioritätszahl

Frage zu Kapitel 4.4

Welchen Aussagen stimmen Sie zu?

- ☐ Der FMEA-Prozess beginnt mit der Auflistung der Funktionalität.
- ☐ Die „6-3-5-Methode“ beschreibt eine Diskussionsform zur Ideenfindung.
- ☐ Zusätzliche Maßnahmen lassen sich unter anderem aus der Entdeckungswahrscheinlichkeit von Fehlern ableiten.
- ☐ Der Initiator einer FMEA kann auch die Moderation übernehmen.

§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

Risikoprioritätszahl (RPZ)

- Die Risikobeurteilung bei einer FMEA erfolgt im Rahmen der Maßnahmenanalyse über die Risikoprioritätszahl (RPZ)
- Für jede Fehlerart eines Betrachtungsgegenstand wird die RPZ ermittelt
- RPZ berechnet sich über die Multiplikation von Einzelbewertungen:
  - Auftretenswahrscheinlichkeit (**A**)
  - Bedeutung bzw. Schwere einer Gefährdung, d.h., einer Fehler-Folge (**B**)
  - Entdeckungswahrscheinlichkeit (**E**)

$$\mathbf{RPZ} = \mathbf{A} * \mathbf{B} * \mathbf{E}$$

- Die Einzelbewertungen werden von der interdisziplinären Experten-Gruppe gemäß genormten Tabellenvorlagen bestimmt

Begriff Auftretenswahrscheinlichkeit (A)

- Bewertung der Auftretenswahrscheinlichkeit (A) einer Fehler-Ursache erfolgt unter Berücksichtigung der definierten Vermeidungsmaßnahmen
- Ist es nahezu sicher, dass eine Fehler-Ursache auftritt, dann entspricht das einer Auftretenswahrscheinlichkeit von 10
- Ist es äußerst unwahrscheinlich, so entspricht das dem Wert 1
- Je detaillierter die Fehleranalyse durchgeführt wurde, desto differenzierter kann die Bewertung erfolgen
- Es wird bei der Bewertung von Ursachen auf bekannte Erfahrungswerte zurückgegriffen (Schadenstatistiken, Zuverlässigkeitsraten,…)

Berechnung der Auftretenswahrscheinlichkeit

$A = (w \cdot g) + f$	
w	**Fehlerwahrscheinlichkeit**
1	Fehlverhalten wird selten erwartet
2	Fehlverhalten wird mit mäßiger Häufigkeit erwartet
3	Fehlverhalten wird sehr häufig erwartet
g	**Gefährdungsdisposition**
1	Aufenthalt im Gefahrenbereich sehr selten
2	Nur zeitweiser Aufenthalt im Gefahrenbereich
3	Sehr langer oder ständiger Aufenthalt im Gefahrenbereich
f	**Anfälligkeit der Gefährdung**
0	Nicht anfällig (gute persönliche Schutzausrüstung)
1	Sehr anfällig (keine Schutzausrüstung)

Bewertungskriterien der Auftretenswahrscheinlichkeit

Allgemeine Bewertungskriterien	Häufigkeit	Bewertungs-punkte
Hoch Es ist nahezu sicher, dass Fehler in größerem Umfang auftreten werden.	1/10 1/20	10 9
Mäßig Mit früherem Fertigungsverfahren vergleichbar, das oft zu Fehlern führte (Prozess wird beherrscht).	1/50 1/100	8 7
Gering Mit früherem Fertigungsverfahren vergleichbar, das gelegentlich, jedoch nicht in einem wesentlichen Umfang, Fehler aufweist (Prozess wird beherrscht).	1/200 1/500 1/1000	6 5 4
Sehr gering Der Prozess wird statistisch beherrscht.	1/2000 1/20 000	3 2
Unwahrscheinlich Die Prozessfähigkeit ist sichergestellt.	≈0	1

Begriff Bedeutung der Fehler-Folge (B)

- Bedeutung der Fehler-Folge (B) bewertet die Folgen einer Fehler-Ursache auf das System
- Unter Folgen sind konkrete Risiken und Gefährdungen zu verstehen
- Bewertung erfolgt stets aus der Sicht des Nutzers bzw. Endverbrauchers
- Wert 10 steht für eine sehr hohe, 1 für eine sehr geringe Gefährdung
- Gleiche Fehler-Folgen müssen stets mit der gleichen Schwere bzw. dem gleichen Zahlenwert bewertet werden

Berechnung der Bedeutung der Fehler-Folge

$B = (v \cdot d) + b$	
v	**Verletzungsgrad**
1	Leichte Verletzungen (Erste-Hilfe-Versorgung)
2	Mittelschwere Verletzungen (ambulante Behandlung notwendig)
3	Sehr schwere Verletzungen (stationäre Behandlung notwendig)
d	**Schadensdauer**
1	Keine Langzeitschäden oder Verletzungsfolgen
2	Noch tragbare Langzeitschäden
3	Schwere Langzeitschäden (Berufsunfähigkeit, Invalidität)
b	**Rettungschancen und Schadensbegrenzung**
0	Gute Rettungschancen, erfolgversprechende Schadensbegrenzung
1	Schlechte Voraussetzungen für Rettung und Schadensbegrenzung

Bewertungskriterien der Bedeutung der Fehler-Folge

Allgemeine Bewertungskriterien	Bewertungs-punkte
Es tritt ein äußerst **schwerwiegender Fehler** auf, der darüber hinaus die Sicherheit und / oder die Einhaltung gesetzlicher Vorschriften beeinträchtigt.	10 9
Es tritt ein **schwerer Fehler** auf, der eine Verärgerung beim Kunden auslöst (Fehlfunktionen). Sicherheitsaspekte oder gesetzliche Vorschriften werden hierdurch nicht berührt bzw. treffen hier nicht zu.	8 7
Es tritt ein **mittelschwerer Fehler** auf, der beim Kunden Unzufriedenheit auslöst. Der Kunde fühlt sich dadurch belästigt oder gestört. Der Kunde wird diese Beeinträchtigungen bemerken bzw. wahrnehmen. („Lautsprecher brummt“ oder „Pedalkräfte zu hoch“)	6 5 4
Der **Fehler ist unbedeutend** und der Kunde wird nur geringfügig belästigt. Der Kunde wird nur geringe Beeinträchtigungen am Untersuchungsgegenstand bemerken.	3 2
Es ist unwahrscheinlich, dass der Fehler wahrnehmbare Auswirkungen auf das Verhalten des Untersuchungsgegenstandes haben könnte. Der Kunde wird den **Fehler wahrscheinlich nicht bemerken**.	1

Begriff der Entdeckungswahrscheinlichkeit (E)

- Bewertung der Entdeckungswahrscheinlichkeit (E) einer Fehler-Ursache erfolgt unter Berücksichtigung der definierten Entdeckungsmaßnahmen
- Es werden auch Entdeckungsmaßnahmen berücksichtigt, die lediglich die Fehler-Folge an sich erkennen und somit nur indirekt auf die Fehler-Ursache geschlossen werden kann
- Bewertung von 10 wird vergeben, wenn keine Entdeckungsmaßnahmen definiert wurden und ein Fehler sicher nicht gefunden werden kann
- Wert 1, wenn ein Fehler mit einer sehr hohen Wahrscheinlichkeit gefunden werden kann

Berechnung der Entdeckungswahrscheinlichkeit E

$$E = (q \cdot k) + r$$

q	**Qualifikation der gefährdeten Person**
1	Fachmann
2	unterwiesene Person
3	Laie, nicht unterwiesene Person
k	**Komplexität der Gefährdungssituation**
1	Geringe Komplexität, Situation gut durchschaubar
2	Mittlere Komplexität, Situation noch durchschaubar
3	Hohe Komplexität, Situation kaum durchschaubar
r	**Reaktions-, Eingreif-, Ausweichmöglichkeit**
0	Gute Reaktionsmöglichkeiten
1	Schlechte Reaktionsmöglichkeiten

Bewertungskriterien der Entdeckungswahrscheinlichkeit

Allgemeine Bewertungskriterien	Häufigkeit	Bewertungs-punkte
Unwahrscheinlich Das Merkmal wird nicht geprüft bzw. kann nicht geprüft werden. Verdeckter Fehler, der in der Fertigung oder Montage nicht entdeckt wird.	<90 %	10
Sehr gering Schwerer zu erkennendes Fehlermerkmal. Erkennung durch visuelle oder manuelle 100%-Prüfung möglich.	>90 %	9
Gering Leicht zu erkennendes, messbares Fehlermerkmal. Erkennung durch 100%-Prüfung möglich (automatisiert).	>98 %	6-8
Mäßig Es handelt sich um ein augenscheinliches Fehlermerkmal. Erkennung durch 100%-Prüfung möglich (automatisiert).	>99,7 %	2-5
Hoch Funktioneller Fehler, der bei den nachfolgenden Arbeitsschritten bemerkt wird.	>99,99 %	1

Analyse der Risikoprioritätszahl

– Je nachdem, welcher Wert sich für die RPZ ergibt, sind bestimmte weitere Maßnahmen sinnvoll:

RPZ	Einstufung	Bemerkung/Maßnahmen
1 < RPZ < 100	Akzeptables Restrisiko	Keine Maßnahme erforderlich.
100 < RPZ < 125	Geringes Restrisiko	Zusätzlicher Warnhinweis erforderlich. (Maschine bzw. Benutzerhandbuch)
125 < RPZ < 250	Erhöhtes Restrisiko	Ergänzende, zusätzliche Schutzmaßnahmen erforderlich. (trennende und nicht trennende Schutzeinrichtungen notwendig)
250 < RPZ < 1000	Inakzeptables Restrisiko	Konstruktive Maßnahmen unbedingt erforderlich.

Frage zu Kapitel 4.5

Welchen Aussagen stimmen Sie zu?

☐ Die RPZ ist das Produkt aus **A**uftretenswahrscheinlichkeit, **B**edeutung einer Gefährdung und **E**rfindungsreichtum von Maßnahmen.

☐ Wird ein Fehler gefunden, so ist die Entdeckungswahrscheinlichkeit 10.

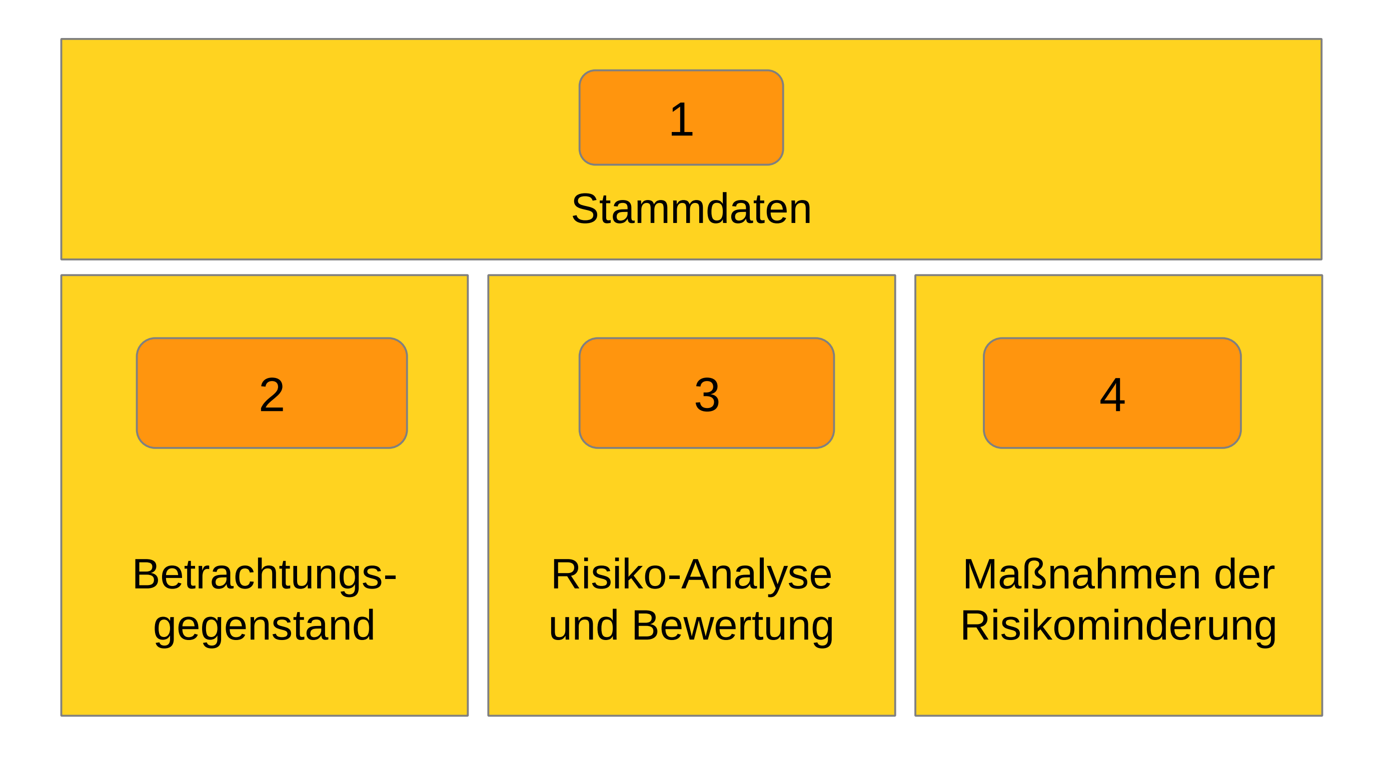
☐ RPZ-Werte ab 250 sind für ein System inakzeptabel.

☐ Akzeptable Risiken erfordern einen Warnhinweis im Benutzerhandbuch.

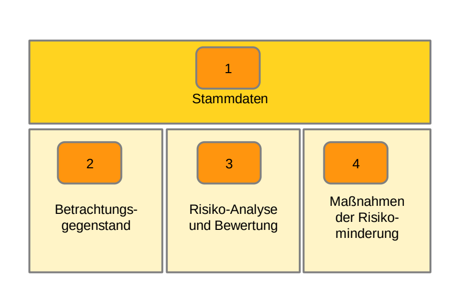
§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

Grundlegendes Schema eines FMEA-Formblatts

- Durchführung und Ergebnis einer FMEA werden in einem Formblatt dargestellt
- Formblättere können einen unterschiedlichen Aufbau oder verschiedene Bezeichnungen haben, jedoch ist die Struktur immer über folgenden 4 Bereiche gegeben:

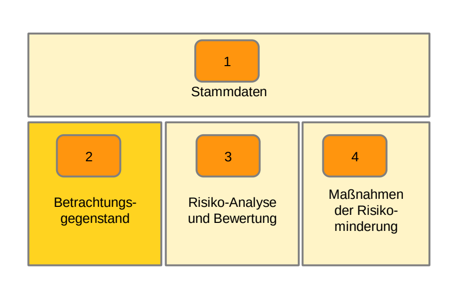


Stammdaten



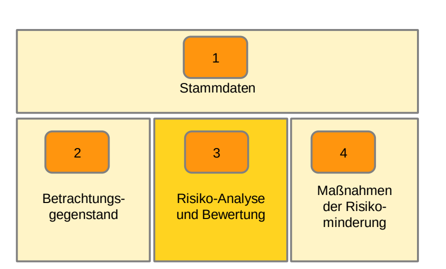
- Im Kopf des FMEA-Formblatts oder auf dem Deckblatt
- Beinhaltet z.B.:
  - Daten des Unternehmens
  - Bearbeiter bzw. FMEA-Team
  - Kenndaten des Projekts
  - Kenndaten des Systemelements (System-Nr.)
- Es muss klar ersichtlich sein:
  - Um welches Projekt es sich handelt
  - Welches Systemelement betrachtet wird
  - Welche Fehler-Folge untersucht wird

Betrachtungsgegenstand



- Enthält:
  - Zu untersuchendes Systemelement
    - Baugruppe
    - Bauteil
    - Prozessschritt
  - Zugehörige Funktionalität
  - Fehlerbeschreibung
    - Fehler-Art
    - Fehler-Ursache
- Daten werden entsprechend den Erfordernissen (Top-, Teil- oder Sub-System) eingetragen

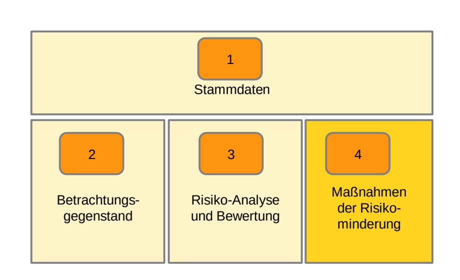
Risiko-Analyse und Bewertung



- Enthält die bestehenden Prüfmaßnahmen:
  - Vermeidungsmaßnahmen
  - Erkennungsmaßnahmen

- Bestimmung bzw. Berechnung der Risikoprioritätszahl (gemäß Kap 4.5):
  - Auftretenswahrscheinlichkeit einer Fehler-Ursache unter Berücksichtigung der Vermeidungsmaßnahmen
  - Bedeutung der Fehler-Folge für das Gesamtsystem
  - Entdeckungswahrscheinlichkeit unter Berücksichtigung aller wirksamen Erkennungsmaßnahmen

Maßnahmen der Risikominderung



- Enthält die, entsprechend der Risikoprioritätszahl, neuen abgeleiteten Maßnahmen zur Fehlerbehandlung
- Verantwortlichkeit bei der Durchführung der Maßnahmen
- Wurden die Maßnahmen durchgeführt, so erfolgt die erneute Bestimmung und Dokumentation der Risikoprioritätszahl im FMEA-Formblatt
- Es wird dadurch ermittelt, ob die getätigten Verbesserungen ausreichend waren, um das Risiko auf einen vertretbaren Rahmen zu mindern oder ob nochmals weitere Optimierungen notwendig sind

Auszug FMEA-Formblatt

Logo / Firma	FMEA	Vorgang	Teilnehmer	Datum

Stammdaten

Prozess-schritt	Funktion	Fehlerart	Fehler-ursache	Derzeitiger Zustand					Empfohlene Abstellungsmaß-nahmen	Verantwortlichkeit/Termin	Verbesserter Zustand				
				Prüfmaß-nahme	A	B	E	RPZ			Maßnahmen	A	B	E	RPZ

Betrachtungs-gegenstand

Risiko-Analyse und Bewertung

Maßnahmen der Risikominderung

Frage zu Kapitel 4.6

Welchen Aussagen stimmen Sie zu?

☐ Stammdaten sind Teil des FMEA-Formblatts.

☐ Betrachtungsgegenstand ist Teil des FMEA-Formblatts.

☐ Instrumente bzw. Werkzeuge der Risiko-Analyse sind Teil des FMEA-Formblatts.

☐ Maßnahmen der Risikominderung sind Teil des FMEA-Formblatts.

§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

4.1 Grundlagen der FMEA

4.2 Arten der FMEA

4.3 Einsatz von Werkzeugen

4.4 Durchführung der FMEA

4.5 Risikoprioritätszahl (RPZ)

4.6 FMEA-Formblatt

4.7 Software-FMEA (SFMEA)

FMEA in der Softwareentwicklung

- FMEA für Software-Systeme muss an die speziellen Anforderungen von Software angepasst werden (vgl. Kap 5):
  - Software unterliegt nicht dem Verschließ
  - Software beinhaltet direkt bei Inbetriebnahme Fehler
  - usw.
- Entscheidend ist die Tatsache, dass Software keine spezifischen, bekannten und eindeutigen Ausfallwahrscheinlichkeiten aufweist
- Eine Bewertung über Wahrscheinlichkeiten fällt somit weg
- Man spricht daher im Zusammenhang mit der Software-FMEA (SFMEA) von einer „Analyse von Möglichkeiten“

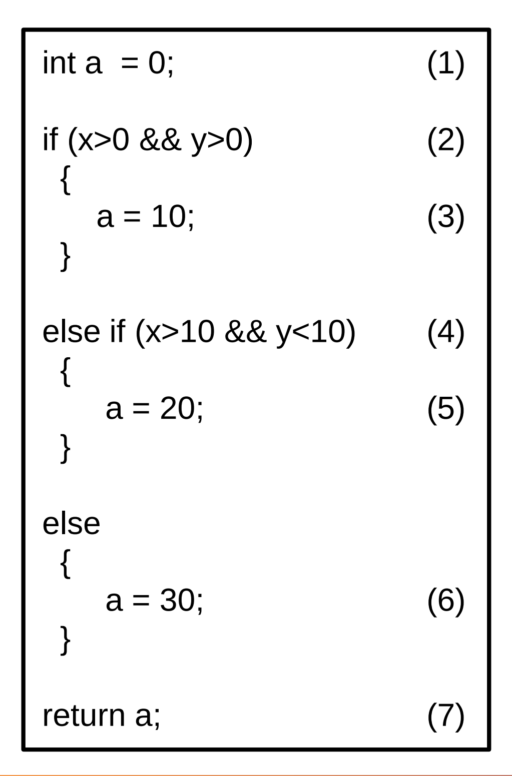
Vorgehen

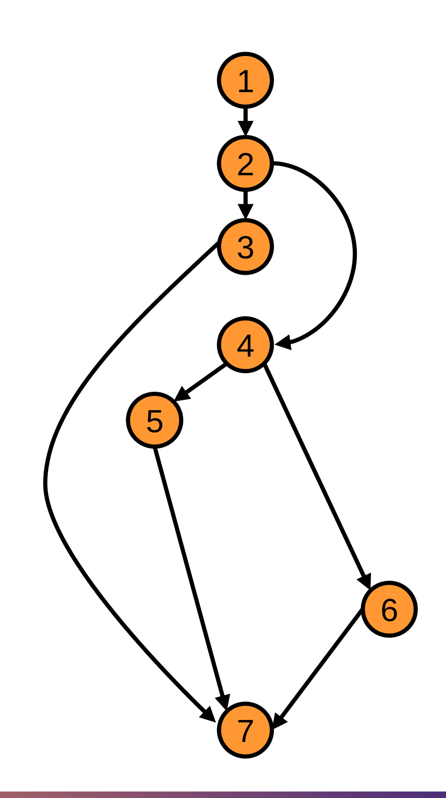
- Es fehlen (noch) allgemeingültige Standards bzw. Prozeduren zur Durchführung einer SFMEA
- Es wird sich daher auf den bekannten Ablauf bezogen:
  1. Strukturanalyse
  2. Funktionsanalyse
  3. Fehleranalyse
  4. Risikobewertung
  5. Optimierung

- Hier erfolgt eine Betrachtung der Schritte 1 - 3, da sich das Kapitel 5 ausführlich mit der Bewertung von Softwarezuverlässigkeit und Maßnahmen zur Steigerung dieser beschäftigt

Strukturanalyse

- Strukturanalyse entspricht prinzipiell der Analyse des Programmablaufs
- Programmablauf wird dem im Entwurf definierten Flussdiagramm entnommen
- Sollte das Flussdiagramm nicht aktuell sein oder fehlen, so kann ein Kontrollflussgraph zu einem Software-Segment erstellt werden
- Beispiel:





Funktionsanalyse

– Die exakte Funktionalität des Systems zu verstehen ist der wichtigste, aber auch komplexeste Schritt
– Anforderungen an das Software-System müssen mit dem zu analysierenden Code abgeglichen werden
  - Überführung von menschlicher Sprache in maschinelle Sprache (Code) ist immer kritisch
  - Fehler bei diesem Schritt führen zu unbrauchbaren Ergebnissen der SFMEA
– Weitere Aspekt ist der Durchführungszeitpunkt:
  - In frühen Phasen liegen oft nur Teile der Gesamtsoftware vor
  - Betrachtung des Gesamtsystems ist nicht möglich

Fehleranalyse (1/2)

- Ansatz der Fehleranalyse ist die Annahme, dass jede Funktion in einen Fehlerzustand geraten kann:
  - Variable wird falsch zugewiesen
  - Eingabewert ist falsch
  - Ungewollter Aufruf eines Moduls
- Es müssen daher alle möglichen Fehler-Ursachen betrachtet werden:
  - Algorithmen-Fehler (Formeln, Modelle,…)
  - Bedatungs-Fehler (Festwerte von Konstanten, Grenzwerte von Variablen)
  - Programmierfehler (Semantik)
  - Methodenfehler (Programmiersprache)
  - Designfehler

Fehleranalyse (2/2)

- Um wirklich alle Fehler-Ursachen betrachten zu können, müssen sämtliche Pfade des Kontrollflussgraphen berücksichtigt werden:
  - Sehr zeitaufwendig und unübersichtlich
  - Gefahr Pfade zu „übersehen“ (unbewusst/bewusst)
- Automatisierte Unterstützung notwendig zur:
  - Generierung der großen Menge an Daten
  - Verarbeitung der Datenmenge

- Notwendig, da nur bei Berücksichtigung aller Fehler-Ursachen und –Zustände eine Ableitung von effektiven Gegenmaßnahmen möglich ist

Frage zu Kapitel 4.7

Welchen Aussagen stimmen Sie zu?

- ☐ Die Überführung von menschlicher in maschinelle Sprache ist immer ein kritischer Vorgang.
- ☐ Bei der SFMEA muss nicht jede Zeile-Code analysiert werden, bereits eine Betrachtung der einzelnen Module ist aussagekräftig.
- ☐ SFMEA wird besonders bei sicherheitskritischen Systemen eingesetzt, da der Code vollständig geprüft wird.
- ☐ SFMEA erlaubt eine klare Bestimmung der Ausfallwahrscheinlichkeit eines Moduls.

Gliederung

§ 1 Einführung, Begriffe und Normen

§ 2 Wahrscheinlichkeit und Zuverlässigkeit

§ 3 Fehlerbaumanalyse (FTA)

§ 4 Fehlermöglichkeits- und Einfluss-Analyse (FMEA)

§ 5 Softwarezuverlässigkeit

§ 6 Zuverlässigkeits- und Sicherheitstechnik

§ 7 Übungsaufgaben

§ 5 Softwarezuverlässigkeit

Softwarekrise – Damals!

- Mitte der 1960er überstiegen Softwarekosten erstmals die Hardwarekosten
- Entwicklungs- und Zuverlässigkeitsmethoden waren nicht an die Komplexität und Anforderungen von Software angepasst:
  - Termine konnten nicht gehalten werden
  - Zeitdruck und, in Folge davon, die Zahl der Softwarefehler erhöhten sich
  - Gesamtkosten überstiegen Planung um ein Vielfaches
- Folge war, dass die ersten großen Software-Projekte scheiterten
- Daher Einführung des Begriffs Software-Engineering und Definition von Entwicklungsvorgehen (NATO-Tagung 1968) mit der Erkenntnis:

> „[…]*when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.”*
>
> (Edsger Dijkstra (1930-2002), niederländischer Informatiker, Wegbereiter der strukturierten Programmierung)

Softwarekrise – Heute?!

- Einsatz, Variation und Komplexität von Software wächst rasant
- Softwaregesteuerte Produkte übernehmen immer wichtigere und kritischere Aufgaben in unserem Alltag
- Aber noch viele offene Punkte und Fragen:
  - Führt der weiterhin steigende Entwicklungs- und Kostendruck auch zu einer immer weiter steigenden Zahl von Fehlern?
  - Kann die Qualitätssicherung mit dem Tempo der Entwicklung Schritt halten?
  - Was sind die Folgen von unzureichender Zuverlässigkeit oder möglicher Sicherheitslücken?
- Softwarekrise ist weiterhin ein ungelöstes Problem der Informationstechnik!

Aspekte von Software (1/2)

- Software zur Realisation komplexer Funktionen nimmt eine immer wichtigere Rolle ein

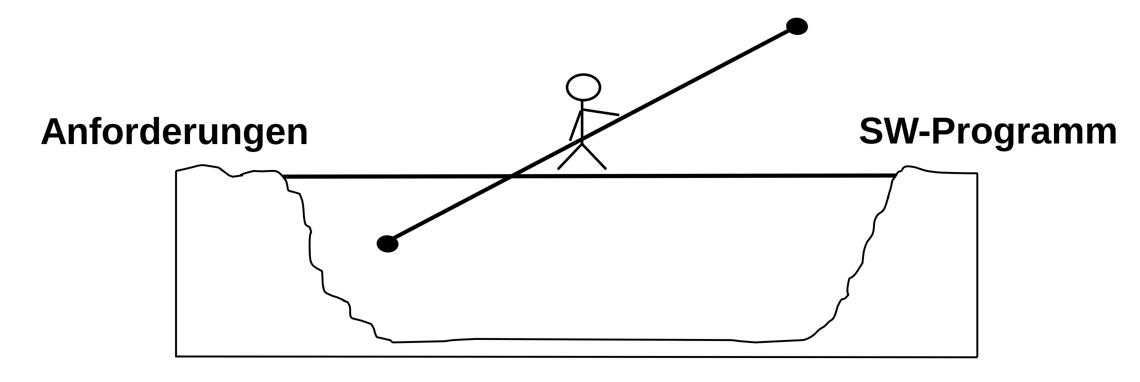
- Jede Software beinhaltet Fehler:
  - Inhärente, systematische Fehler
    - Spezifikationsfehler
    - Entwurfsfehler
    - Implementierungsfehler
    - Dokumentationsfehler
  - Verhalten abhängig vom Betriebsprofil (Umgebung,…)

Aspekte von Software (2/2)

- Testverfahren zeigen die Anwesenheit von Fehlern

- Keine genauen, auf empirischen Untersuchungen beruhenden Daten und Aussagen über die Ausfallrate λ von Software möglich

- Entwicklung zuverlässiger Software ist ein Drahtseilakt



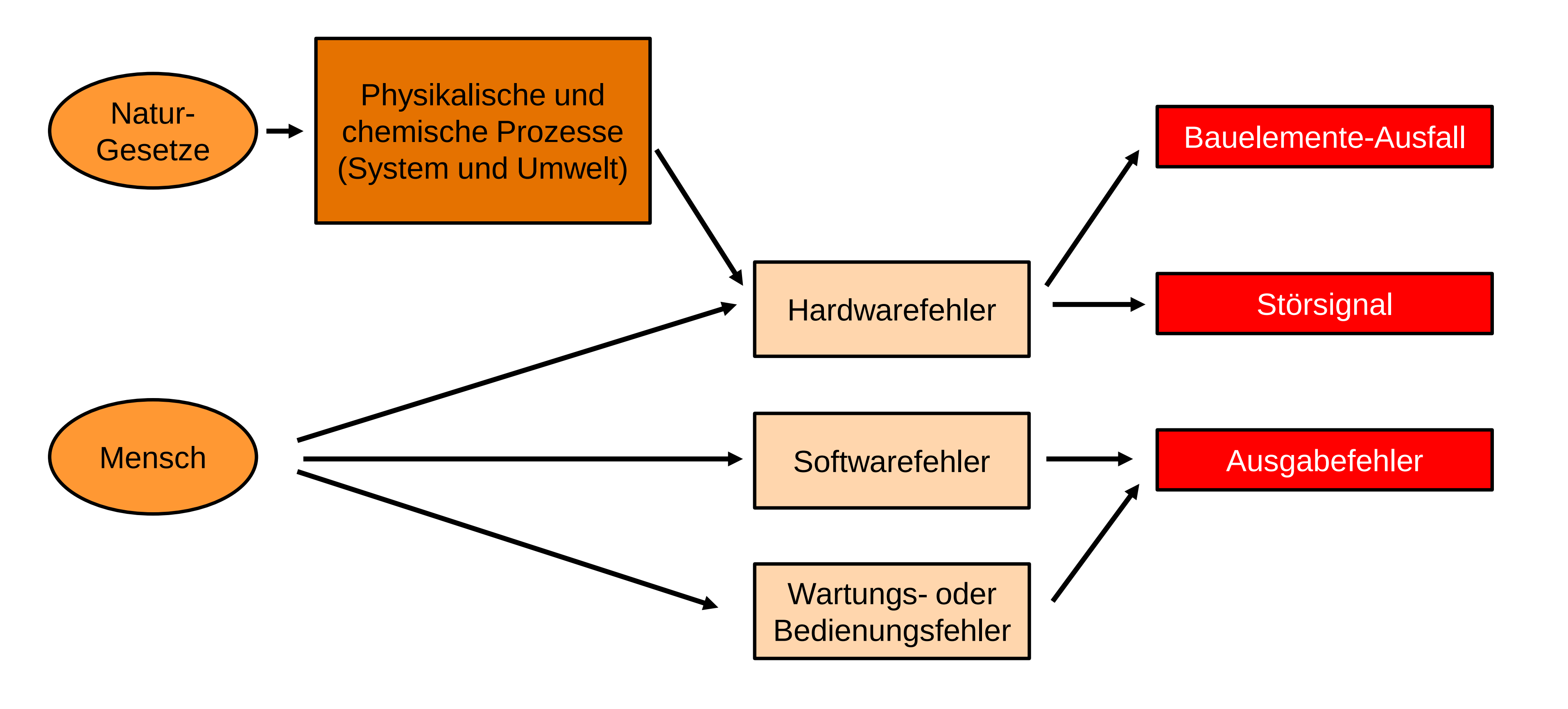
Hardware-Einheiten und Software-Einheiten

– Basis der Software-Zuverlässigkeitsarbeit ist das Verständnis der Unterschiede zwischen Hardware- und Software-Einheiten:

Hardware-Einheiten	Software-Einheiten
Durchlaufen immer mehrere Fertigungsphasen	Werden nach einmaliger Implementierung einfach kopiert
Fehlerfrei bei Inbetriebnahme	Enthält Fehler bei Inbetriebnahme
Ausfall zu einem nicht vorhersagbaren Zeitpunkt	Ausfall unter nicht vorhersagbaren Mechanismen
Ausfall führt zum sofortigen Versagen	Ausfall in Abhängigkeit bestimmter Konstellationen

Fehlerursache von Hardware und Software

– Einordnung und Abgrenzung von Hardware- und Software-Fehlern:



Beispiele für Softwarefehler

– Betrachtet wird ein simples C-Programm mit eingebauten Fehlern:

```
int a;
int c = 10;

int main()
{
        if (a <= c)
        {
                printf("a ist c oder kleiner als c! \n");
                return;
        }

        else if (a = b)
        {
                printf("a ist b! \n");
                return;
        }

        else
        (
                printf("a ist weder b noch c, sowie größer als c! \n");
                return;
        )
}
```

Typische Softwarefehler (1/2)

- Datenreferenz
    - Initialisierungsfehler
    - Indizes außerhalb definierter Grenzen
- Datendeklaration
    - Deklaration der Variablen fehlt
    - Attribute wurden falsch verstanden
- Berechnung
    - Verwendung inkorrekter Datentypen (Rundungsfehler, Überläufe,…)
    - Falsch gesetzte Klammern
- Vergleich
    - Inkorrekte Boolesche Ausdrücke
    - Prioritäten der Vergleichsoperatoren wurden falsch verstanden

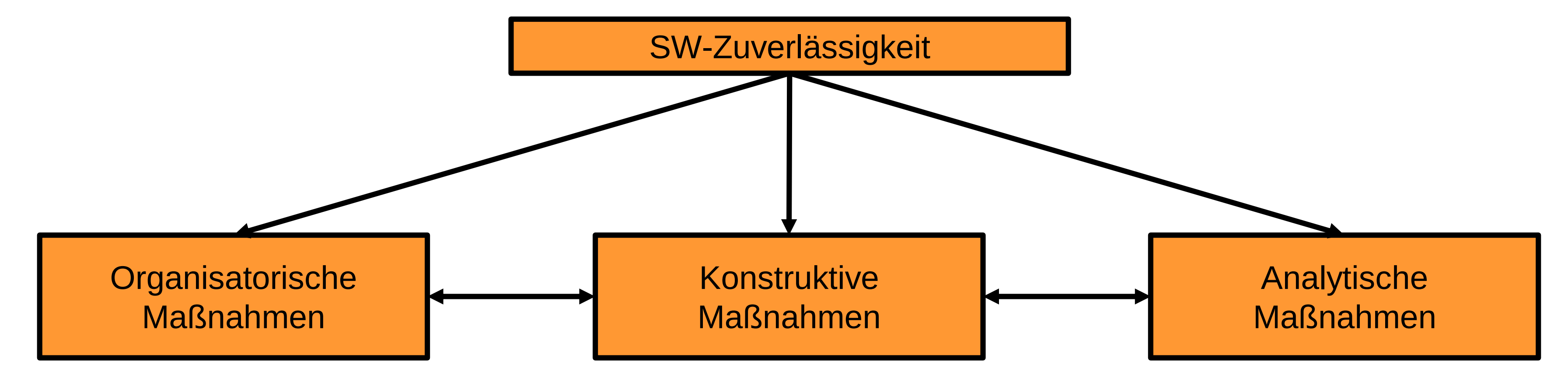
Typische Softwarefehler (2/2)

- Steuerfluss
  - Falsche Schleifenendekriterien
  - Struktur fehlerhaft (z.B. bei DO/END-Anweisung)
- Schnittstellen
  - Verwechslung von Variablen und Konstanten
  - Parameter oder Argumente nicht konsistent
- Ein-/Ausgabe
  - Fehlerhafte Formate der Daten
  - Unklarer Umgang mit Fehlern bei der Ein- und Ausgabe
- Sonstiges
  - Funktionen wurden nicht realisiert
  - Warnungen bei der Kompilierung wurden nicht beachtet

Betrachtung der Softwarezuverlässigkeit

– Eine hohe Zuverlässigkeit von Softwaresystemen wird durch Anwendung und Zusammenspiel verschiedener Maßnahmen-Arten erreicht

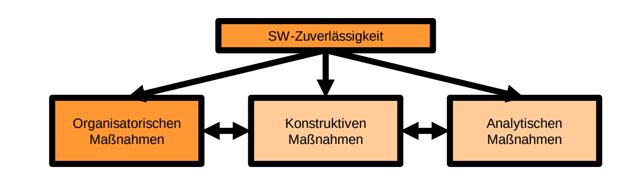
– Es gilt:



– Gegenseitige Beeinflussung, daher:

- Vorausschauendes Planen erspart Arbeit bei der Implementierung!
- Strukturierte Implementierung erspart Arbeit beim Testen!

Organisatorische Maßnahmen



- Organisatorische Maßnahmen zur Erhöhung der Softwarezuverlässigkeit haben das Ziel, definierte und kontrollierbare Prozesse zu gestalten
- Ausarbeitung und Einhaltung von Merkmalen ist Bestandteil der Maßnahmen
- Beispiele für organisatorische Maßnahmen:
  - Entwicklung von Software nach Schema eines Vorgehensmodells
  - Anwendung von Konfigurationsmanagement zur Steuerung und Überwachung aller Änderungen und Aktivitäten
  - Entwicklung nach allgemeingültigen Software-Qualitätsmerkmalen (bereits in der Analyse- und Entwurfsphase)

Qualitätsmerkmale nach ISO/IEC 9126

– Zuverlässigkeit (*reliability*)

„Wird die geforderte Funktion fehlerfrei ausgeführt?“

– Benutzbarkeit (*useability*)

„Ist die Software verständlich und einfach bzw. intuitiv benutzbar?“

– Effizienz *(efficiency)*

„Wird der kleinste Bedarf an Rechenzeit, Speicher und Peripherie erreicht?“

– Änderbarkeit *(changeability)*

„Können Merkmale mit kleinstmöglichem Aufwand geändert werden?“

– Übertragbarkeit *(portability)*

„Ist die Software unabhängig vom Betriebssystem und der Hardware?“

– Funktionalität *(functionality)*

„Tut die Software was sie soll?“

Weitere Qualitätsmerkmale von Software

– Dokumentation *(documentation)*

„Liegen alle erforderlichen Dokumente vor und sind diese aktuell?“

– Integrität *(integrity)*

„Ist die Software sicher gegen eine unerlaubte Anwendung?“

– Konsistenz *(consistency)*

„Enthält die Software eindeutige Zuordnungen?“

– Robustheit *(robustness)*

„Ist die Software stabil bei Fehlbedienung oder Überlast?“

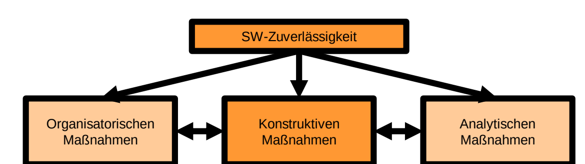
– Prüfbarkeit *(testability)*

„Kann die Software leicht beurteilt bzw. geprüft werden?“

– Vollständigkeit (completeness)

„Ist es nachgewiesen, dass die gesamte geforderte Funktionalität erfüllt ist?“

Konstruktive Maßnahmen



– Konstruktive Maßnahmen zur Erhöhung der Softwarezuverlässigkeit haben das Ziel, eine strukturierte Vorgehensweise in der Entwicklung zu ermöglichen

– Konstruktive Maßnahmen dienen der Gestaltung der Zuverlässigkeit und sollen präventiv die Entstehung von Fehlern vermeiden

– Einsatz von:
  - Richtlinien und Prinzipien für den Entwicklungsprozess
  - Methoden und Techniken zur Programmierung
  - Einsatz von Prototypen
  - Verwendung von Werkzeuge

Anwendung konstruktiver Techniken (1/2)

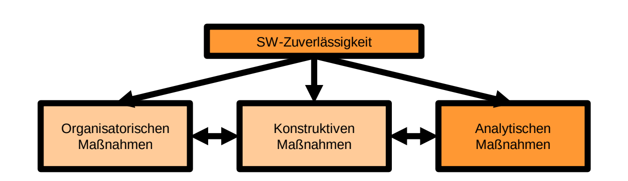
– Gegebene Anforderungen müssen genau überprüft werden, nach :
  - Strukturiertheit
  - Erkennbarkeit und Nachprüfbarkeit
  - Vollständigkeit
  - Widerspruchsfreiheit und
  - Verständlichkeit

– Es müssen Absicherung eingeplant werden:
  - gegen typische Programmierfehler
  - jede Art der Fehlbedienung und
  - mögliche äußere Störfaktoren

Anwendung konstruktiver Techniken (2/2)

– Bei der Implementierung ist zu beachten:
  - Vorgehen nach Top-Down-Methode
  - Verwendung höherer Programmiersprachen
  - Strukturierung und Trennung von Funktionen (Modularität)
  - Format der Daten
  - Aspekte der Prüfbarkeit
  - Klarer Programmierstil
  - Geeignete Verwendung von Kommentaren

Analytische Maßnahmen



– Analytische Maßnahmen zur Erhöhung der Softwarezuverlässigkeit haben das Ziel, möglichst viele in der Software enthaltenen Fehler und Mängel zu entdecken

– Es werden gezielt Informationen gesammelt, um den Zustand eines Prüfobjekts zu bewerten

– Effektives und erfolgsversprechendes Prüfen ist schwierig, da bereits einfache Programme eine große Anzahl zu beachtender Zustände aufweisen

– Daher sorgfältige Planung und Definition von:
  - Aufgaben
  - Zielen
  - Strategien bei der Ausführung

Prüfgrundsätze

- Prüfung kann nur gegen Vorgaben erfolgen (Anforderungen, Vergleiche,…)
- Prüfung muss schon während des Entwurfs geplant werden
- Verwendung geeigneter Hilfsmittel (Debugger, Testgeneratoren,…)
- Prüfverfahren müssen reproduzierbare Ergebnisse liefern
- Ergebnisse müssen dokumentiert werden, wobei jeder Fehler ausreichend protokolliert wird:
  - Angabe der Prüfdauer bis zur Entdeckung
  - Ursache, Art und Folge des Fehlers
  - Aufwand der Fehlerbeseitigung
- Bei komplexen Softwaresysteme kann die Prüfung mit zugehöriger Fehlerkorrektur über 50% des Entwicklungsaufwand beanspruchen

Konsequenzen der Maßnahmen für den Entwicklungsprozess

- Bei der Entwicklung von Software müssen Fehlfunktionen nicht nur erkannt und korrigiert werden, sondern die Software selbst muss robust gegen das Auftreten von Fehlfunktionen sein!
- Alle Entscheidungen und Änderungen am Entwurf müssen ausführlich dokumentiert werden!
- Es müssen feste Verantwortlichkeiten vergeben werden!
- Prüfung von Software ist die einzige praktische Möglichkeit um Fehler zu entdecken und somit die Zuverlässigkeit zu erhöhen!
- Es muss bei der Wiederverwendung von Software genau geprüft werden, ob die Einsatz- bzw. Umgebungsbedingungen übereinstimmen!

Frage zu Kapitel 5.1

Welchen Aussagen stimmen Sie zu?

- ☐ Tests weisen die Fehlerfreiheit von Software nach.
- ☐ Ein Softwarefehler führt nicht immer zum Ausfall eines Systems.
- ☐ Menschliche Irrtümer sind der Grund für Fehler in Software.
- ☐ Kommentare in Software haben keinen Einfluss auf deren Zuverlässigkeit.

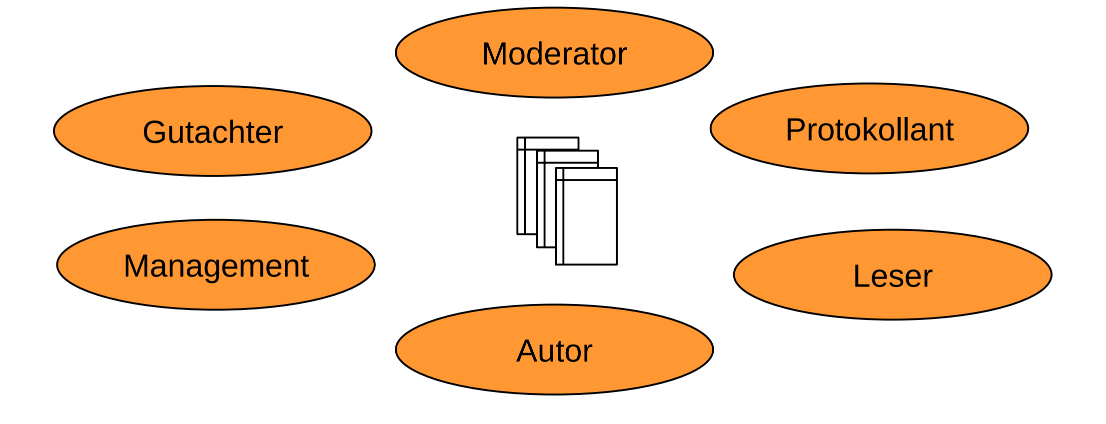
§ 5 Softwarezuverlässigkeit

Unterscheidung von Maßnahmen

- Maßnahmen, um die Entstehung von Softwarefehler über einen verbesserten Entwicklungsprozess möglichst zu vermeiden
  - Reviews
  - Formale Anforderungsspezifikation
- Maßnahmen, um vor der Inbetriebnahme möglichst viele der enthaltenen Softwarefehler über ein sicherheitsbezogenes Vorgehensmodell aufzudecken
  - Testen
  - Diversitäre Rückwärtsanalyse
- Maßnahmen, um gefährliche Auswirkungen von Softwarefehler während des Betriebs zu verhindern
  - Redundanz
  - Software-Diversität

Review

- Review ist nach IEEE 729 ein mehr oder weniger formal geplanter und strukturierter Prozess zur Analyse und Bewertung schriftlicher Dokumente
- Schema:



- Ziel:
  - Feststellung von Mängeln, Fehlern und Unvollständigkeiten
    - Fehler im Design
    - Falsche Schnittstellenspezifikation
    - Unzureichende Wartbarkeit
  - Formale Planung und Strukturierung des Bewertungsprozess
  - Formale Abnahme des Prüfobjekts

Allgemeine Review-Methoden

- Entwurfs- oder Programmreviews
  - Aufdecken von Fehlern im Entwurf oder Programm
  - Durchführung anhand von Checklisten
- Fortschrittsreview
  - Bereitstellung von Informationen über den Fortschritt des Projekts für das Management
  - Betrifft vor allem Kosten und Termine
- Qualitätsreview
  - Technische Analyse des Produkts (Komponenten und Dokumente)
  - Aufdeckung von Fehlern und Inkonsistenzen zwischen der Systemspezifikation, dem Entwurf, dem Programm und der Dokumentation

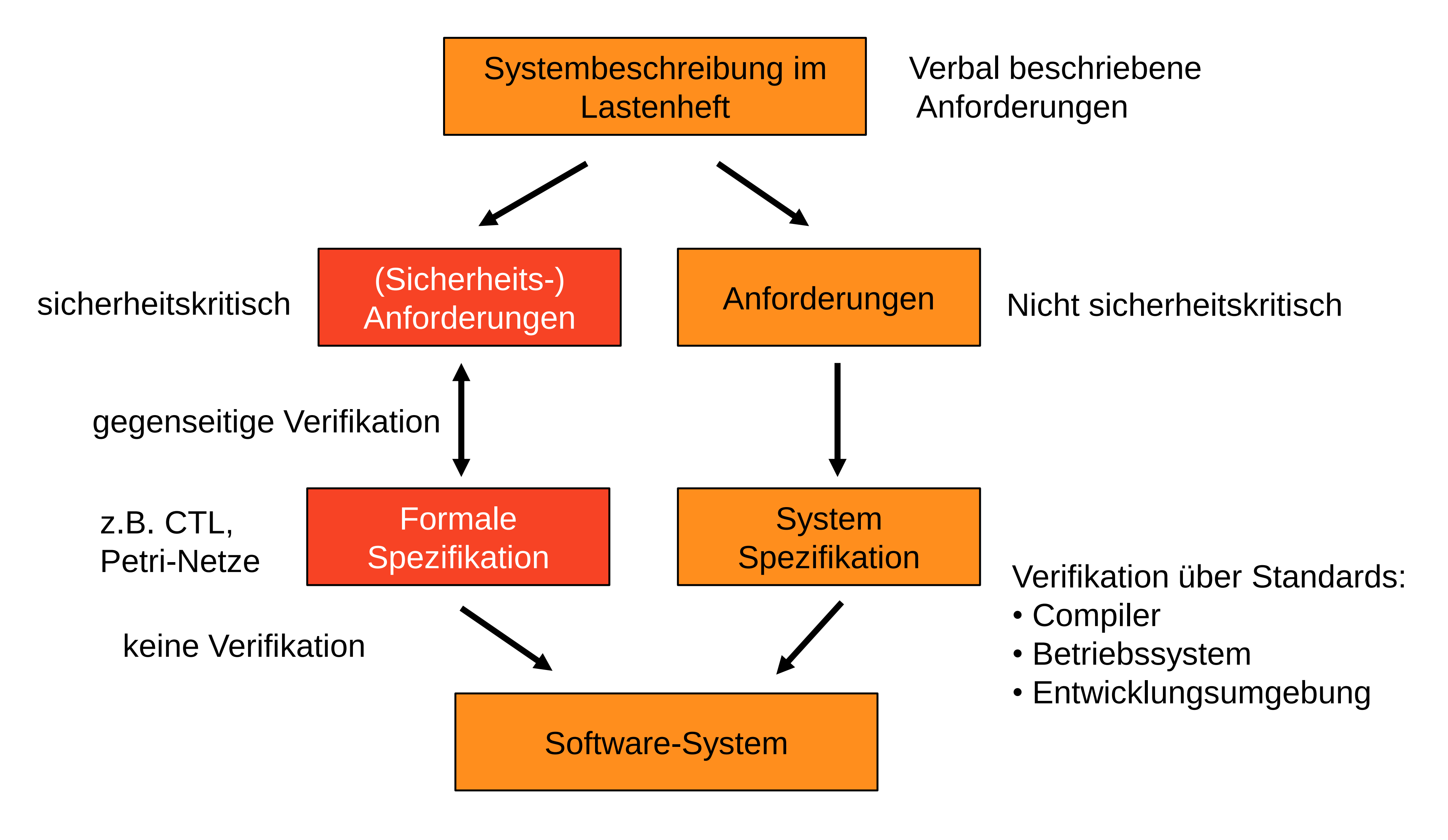
Review-Arten nach IEEE 1028

- Management-Review zur Bewertung des Projektstands durch das Management
- Technisches-Review zur Bewertung des Produkts durch Gutachter
- Walk-Through zur komprimierten Vorstellung des Produkts durch einen Autor
- Inspektion zur Betrachtung von Fehlern durch Gutachter
- Audit zum Nachweis von Konformität durch unabhängige Dritte (Leser und Protokollant)

Formale Spezifikation von Anforderungen

- Formale Spezifikation ist die Beschreibung von Software bzw. eines Programms mittels einer formalen Sprache
- Formale Sprache entspricht einer eindeutig definierten Notation über Zeichen, Wörter, Symbole oder Phrasen
- Ziel ist die präzise Beschreibung der zu lösenden Aufgabe bzw. des zu analysierenden Systems
- Es erfolgt der Einsatz von Werkzeugen zur Überprüfung der Regeleinhaltung der formalen Sprache
- Ermöglicht die Prüfung des Softwaresystems gegen die Anforderungen, d.h., gegen die Systembeschreibung im Lastenheft

Ablauf formale Anforderungsspezifikation



Bewertung einer formalen Spezifikation

- Stärken:
  - Immer eindeutig, da Bedeutung der Notation definiert
  - Neutrale und widerspruchsfreie Lösung
  - Erfüllung von Eigenschaften (Sicherheitsanforderungen) beweisbar
  - Modelle sind simulierbar (z.B. Petri-Netze)
- Schwächen:
  - Sehr aufwendige Durchführung
    - Zerlegung des Systems
    - Beschreibung von Ausnahmefällen
  - Erstellung und Prüfung nur über ausgebildetes Fachpersonal
  - Teilweise einseitige Ausrichtung, z.B. bezogen auf:
    - Die Funktionalität (mathematische Notationen)
    - Das Verhalten (Petri-Netze)

Prozess des Testens

- Testen ist ein Verfahren bzw. ein Versuch der Softwareprüfung mit dem Ziel, ein Programm so auszuführen, dass Fehler gefunden werden
- Vorgehen:

Testvorbereitung (TV):

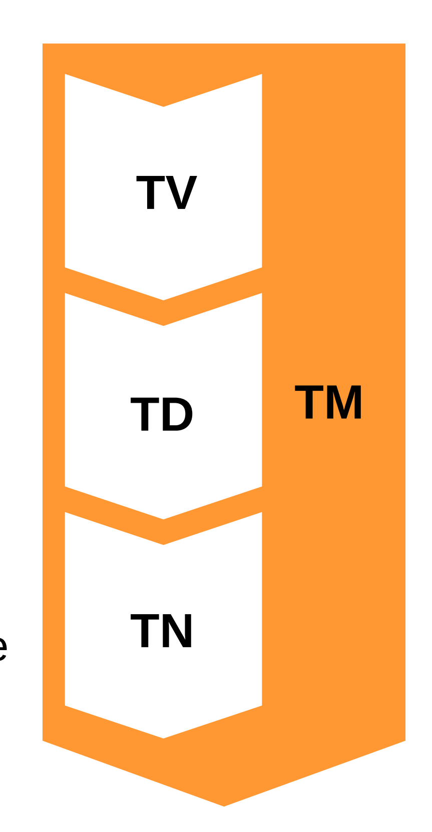
- Ermittlung von Testfällen
- Spezifikation

Testdurchführung(TD):

- Ausführung der Testfälle
- Protokollierung

Testnachbereitung (TN):

- Aufbereitung der Testergebnisse
- Bewertung

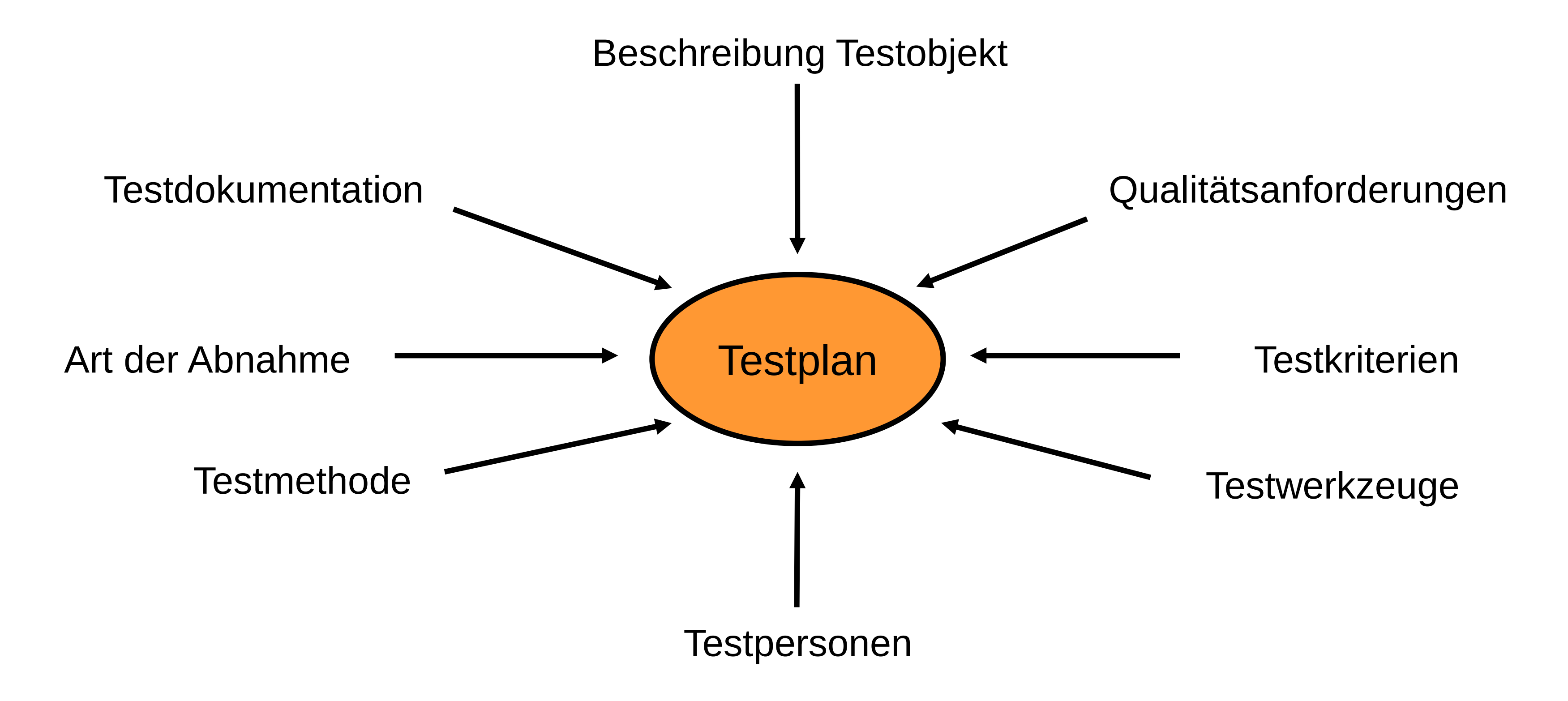


Testmanagement (TM):

- Planung des Testprozess
- Steuerung
- Kontrolle

Testplanung

- Testplanung ist die Grundlage der Qualitätssicherung und Voraussetzung für die Überwachung und Steuerung der Testausführung
- Testplan beinhaltet Aufgaben, Ziele und Strategien der Testausführung:



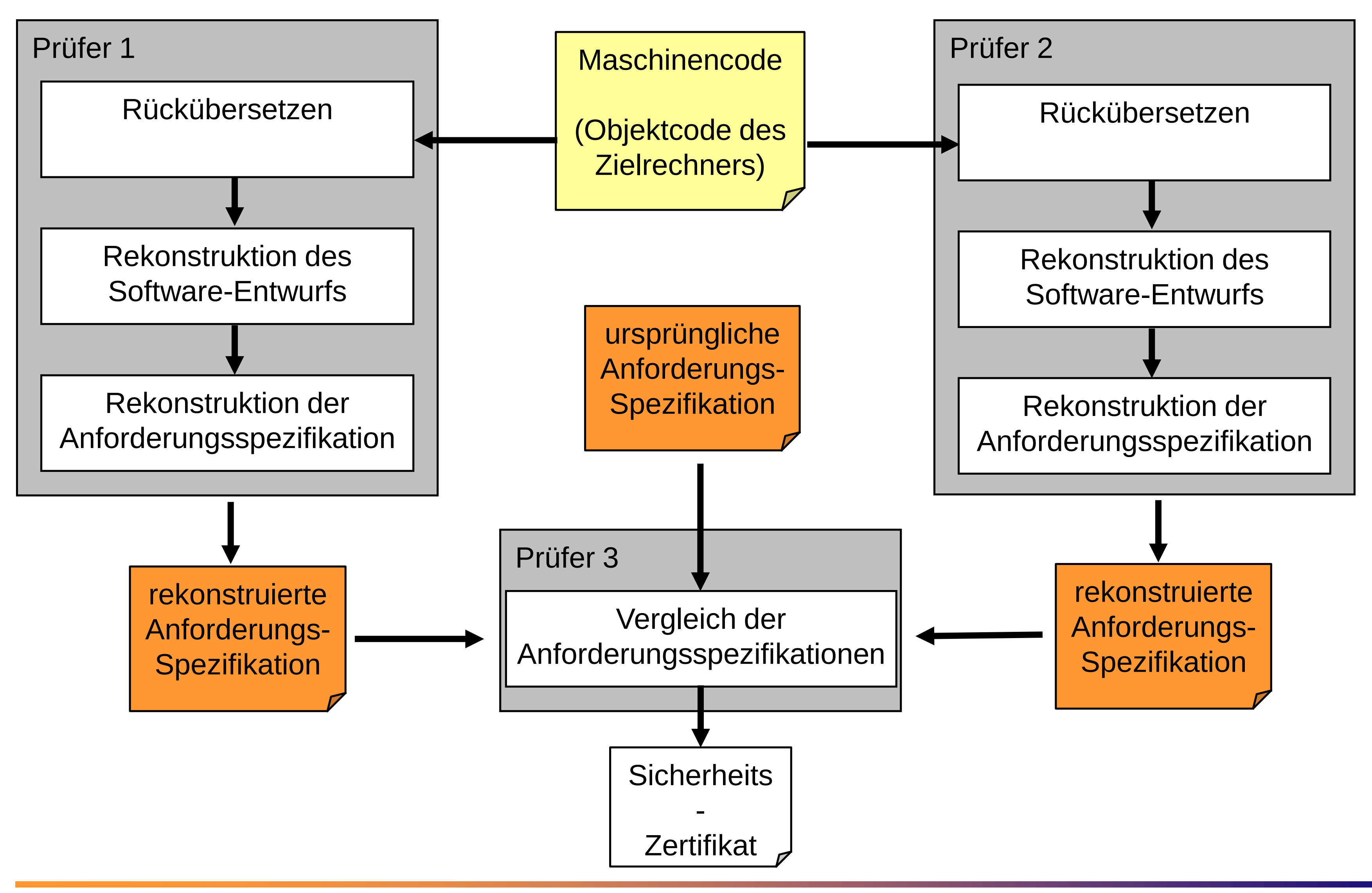
Hinweise zum Testen

- Ein modularer Aufbau des Testobjekts (mit definierten Schnittstellen) erleichtert die Fehlerlokalisierung über:
  - Programmablaufverfolgung
  - Zustandsverfolgung
  - Zustandsüberwachung
- Sorgfältige Durchführung der Fehlerkorrektur zur Vermeidung von:
  - Neuen Fehlern
  - Mehrfacher Wiederholung der Testfälle
- Beachtung bekannter, typischer Fehler z.B.:
  - Endlosschleifen
  - Falsche logische Operationen
  - Sonderfälle

Diversitäre Rückwärtsanalyse

- Vom TÜV Rheinland, im Rahmen eines Projekts am experimentellen Kernkraftwerk in Norwegen, entwickelte Methode zur Software-Verifikation und zum Sicherheitsnachweis von Software-Systemen
- Prinzip:
  - Objekt wird aus der Zielmaschine ausgelesen und an zwei Prüfgruppen übergeben
  - Prüfgruppen arbeiten in Folge ohne Kontakt untereinander
  - Code wird rückübersetzt bis hin zu den spezifizierten Anforderungen
  - Rückspezifikationen werden untereinander und mit den ursprünglichen Anforderungen verglichen
  - Bei Übereinstimmung gilt Softwaresystem als korrekt entwickelt und erhält Sicherheitsnachweis
  - Annahme: System ist frei von Fehlern nach dem Stand der Technik

Ablauf der diversitären Rückwärtsanalyse



Bewertung

- Vorteile:
  - Leicht verständlich und direkt anwendbar
  - Sehr wirkungsvoll, da alle Abweichungen aufgedeckt werden
- Nachteile:
  - Großer Aufwand
  - Keine Betrachtung möglicher Fehlerfälle während des Betriebs

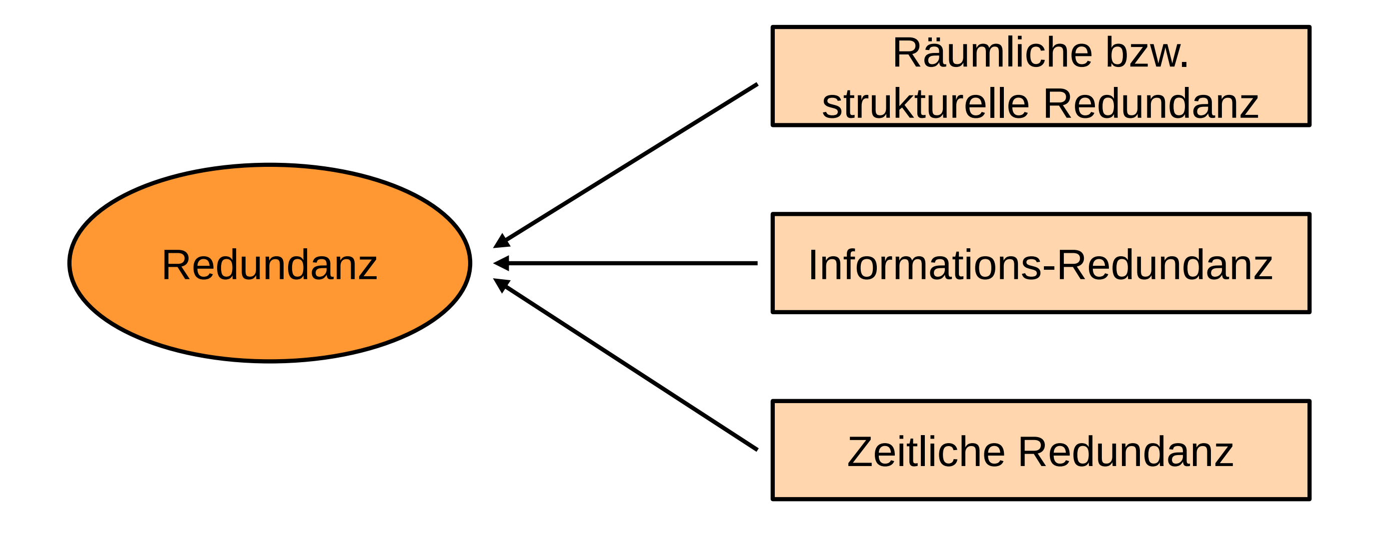
- Verringerung des Aufwands durch Rechnerunterstützung:
  - Im ersten Zweig erfolgt Überprüfung durch einen Prüfer bzw. Prüfgruppe
  - Im zweiten Zweig wird ein speziell entwickeltes Werkzeugsystem zum Rückübersetzung eingesetzt
  - Prüfer bzw. Prüfgruppe vergleicht die automatisierten Ergebnisse mit den Eigenen

Redundanz

– Redundanz bedeutet das Vorhandensein zusätzlicher, funktionsbereiter Mittel

– Methode, um gefährliche Auswirkungen verbleibender Restfehler während des Betriebs zu verhindern

– Kategorisierung:



Redundanz-Kategorien (1/2)

- Räumliche bzw. strukturelle Redundanz
  - Wird häufig auch als Hardware-Redundanz bezeichnet
  - Aber alle Techniken können auf Software übertragen und implementiert werden

Redundanzform	Merkmale
Heiße Redundanz	Mehrere Systeme (mind. 3), die gleichzeitig die gleiche Funktion ausführen.
Kalte Redundanz	Mehrere Systeme, die eine gleiche Funktion ausführen können.
Standbye-Redundanz	Zusätzliche Mittel im System, auf die im Fehlerfall umgeschaltet wird (Komponenten, kein ganzes System).
N+1-Redundanz	N aktive Einheiten und 1 passive Einheit im System, die im Fehlerfall eine aktive Einheit ersetzen kann

Redundanz-Kategorien (2/2)

– Informations-Redundanz
  - Einsatz, um Fehler bei der Kommunikation zwischen Komponenten zu vermeiden bzw. zu reduzieren oder Speicher gegen Fehler abzusichern
  - Kodierungs-Techniken und –Verfahren

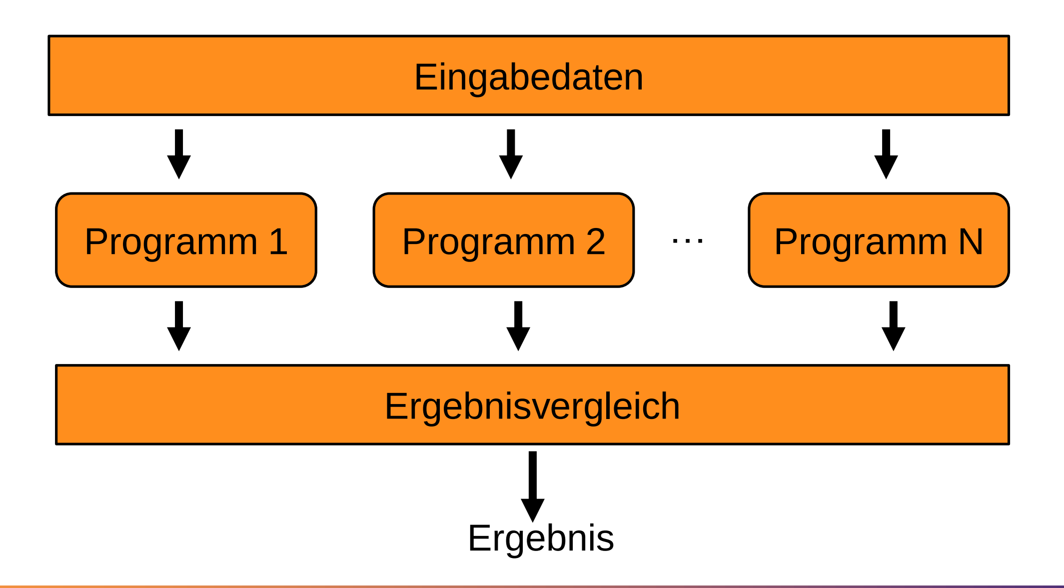
– Zeitliche Redundanz
  - Anweisungsausführung erfolgt zu unterschiedlichen Zeiten
  - Informationen der Ausführung werden mit einem Datum versehen
  - Die mit dem jeweiligen Datum versehenen Informationen werden zeitlich getrennt voneinander an die jeweiligen Komponenten übertragen
  - Besonders geeignet, um intermittierende Fehler zu erkennen

Software-Diversität

– Zur Vermeidung der Kopie von Fehlern, wie bei redundanten Systemen, erfolgt eine Realisierung von Gleichartigkeit über unterschiedliche Mittel:

- Unterschiedliche Algorithmen
- Unterschiedliche Programmiersprachen
- Unterschiedliche Automatisierungsgeräte/Steuerungen
- Unterschiedliche Entwicklungs-Teams

– Struktur:



Probleme bei Software-Diversität

- Nachweis der Diversität grundsätzlich problematisch
- Zeitliche Koordinierung diversitärer Programme ist aufwendig und schwierig
- Diversitäre Programme verursachen hohe Kosten
  - Entwicklung
  - Nachweis
  - Wartung und Pflege
- Diversität muss auch nach Änderungen gewährleistet bleiben
- Auswahl der Art (Algorithmus, Programmiersprache,…)
- Denkmuster von Menschen

Frage zu Kapitel 5.2

Welchen Aussagen stimmen Sie zu?

- ☐ Die Überführung einer verbalen in eine formale Sprache ist immer ein kritischer Schritt.
- ☐ Ein Audit ist eine komprimierte Vorstellung eines Produkts.
- ☐ Bei Standbye-Redundanzen erfolgt im Fehlerfall ein Umschalten auf ein separates zweites System.
- ☐ Software-Diversität zeichnet sich durch geringe Kosten aus.

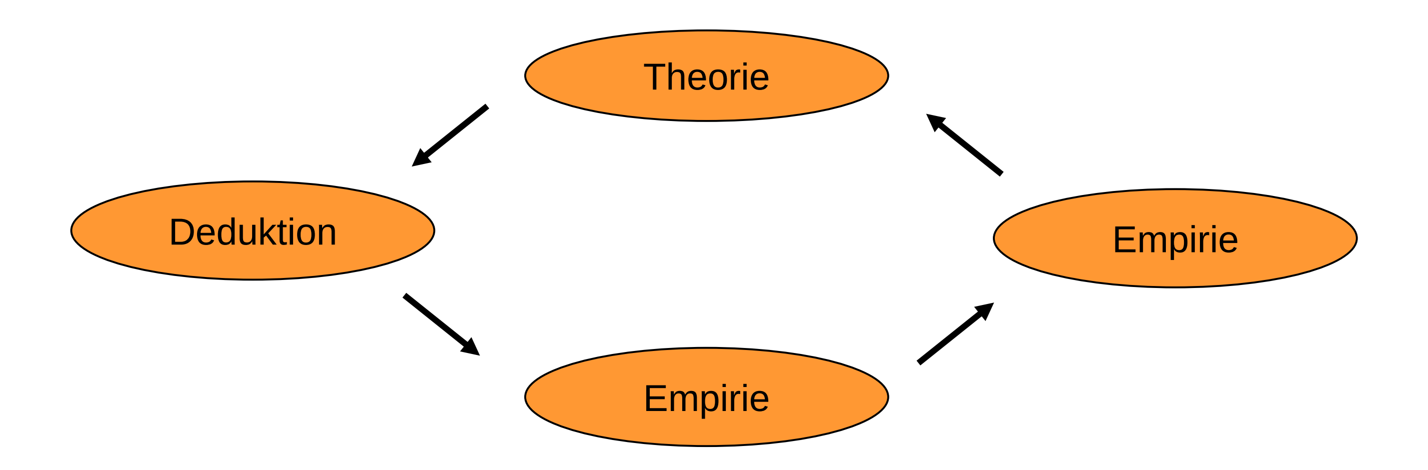
§ 5 Softwarezuverlässigkeit

Software-Zuverlässigkeitsmodelle

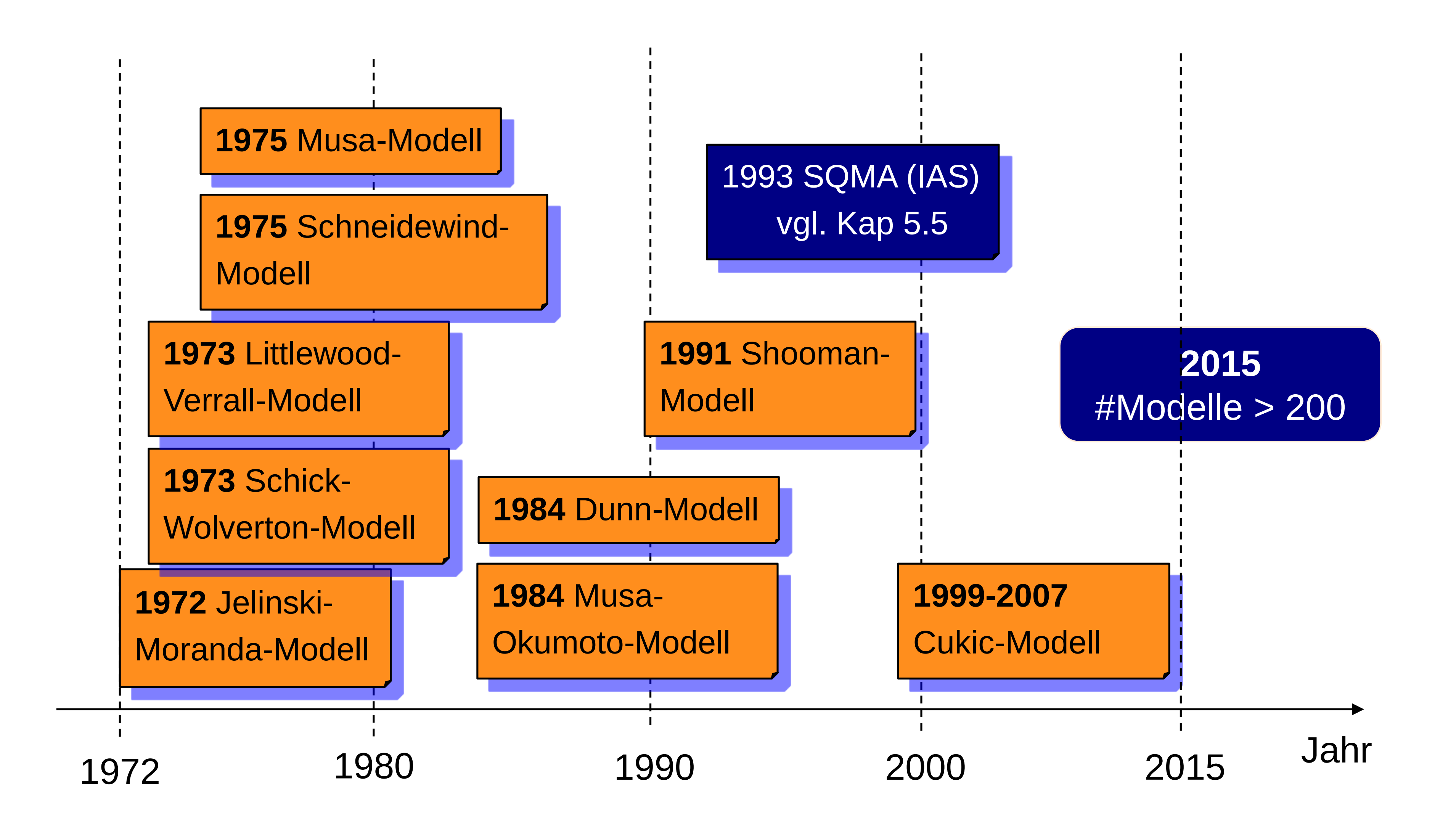
– Zuverlässigkeitsmodelle in der Softwaretechnik sind mathematische Modelle zur:
  - Abschätzung der Zuverlässigkeit einer Software
  - Berechnung von Fehlerraten
  - Schätzung der Anzahl von Rest-Fehlern in einem Codeabschnitt

– Basis aller Modelle sind empirische Daten (z.B. gesammelte Daten von Tests)

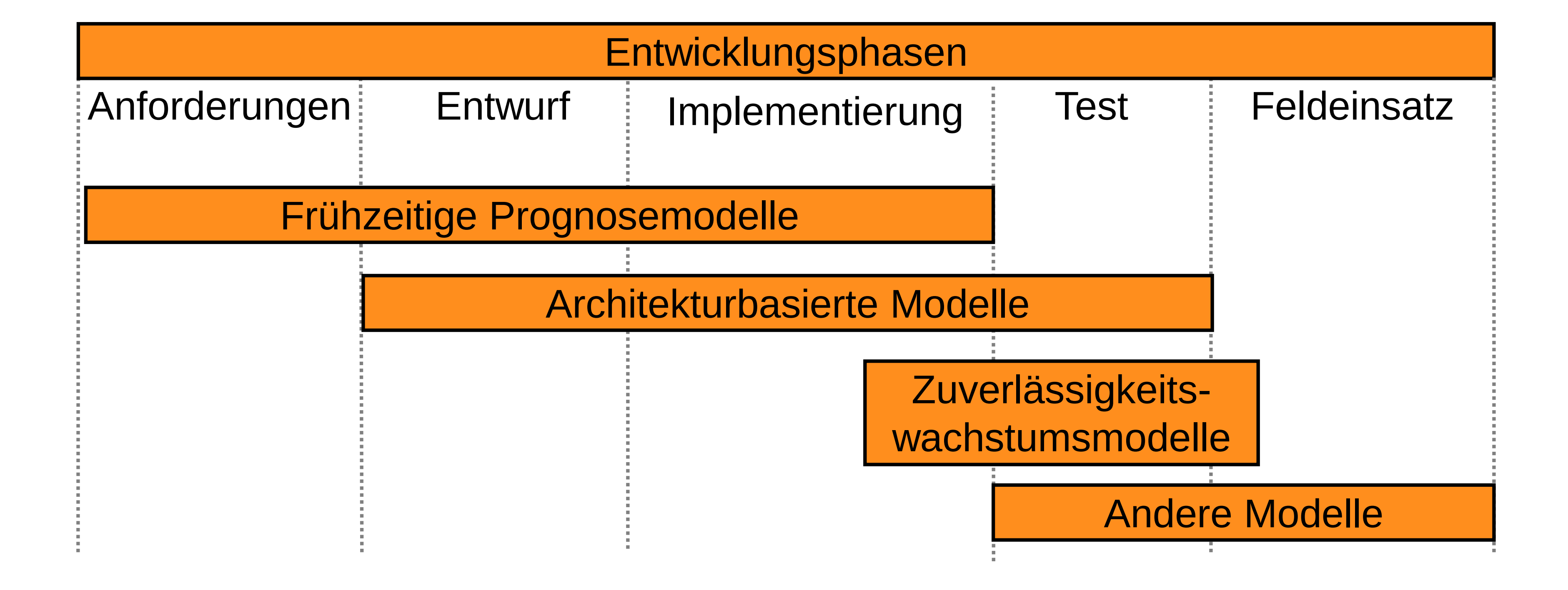
– Erkenntnisse aus empirischen Daten werden als Empirie bezeichnet:



Klassifizierung nach geschichtlich-zeitlichem Auftreten

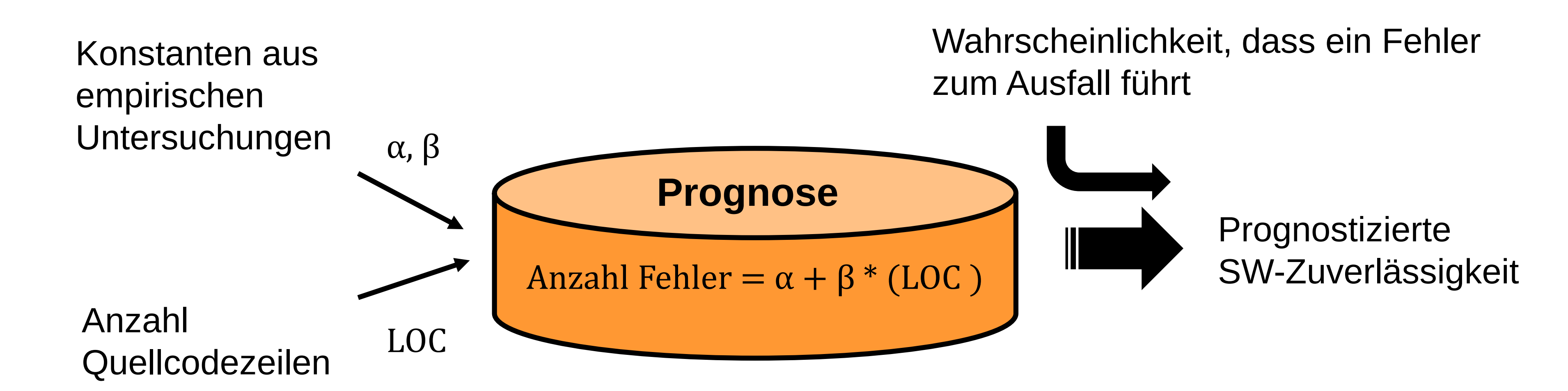


Klassifizierung nach Entwurfsphasen



Frühzeitige Prognosemodelle - Prinzip

- Aus den, je nach Modell benötigten, empirischen Daten von Software lässt sich eine Prognose der Zuverlässigkeit ableiten
- Bereits in frühen Phasen ist eine Abschätzung der Zuverlässigkeit für eine Software möglich
- Prinzip:



Frühzeitige Prognosemodelle - Bewertung

- Problematisch ist die Genauigkeit der Ergebnisse
- Es ist kaum vorhersehbar ob, wie viele und wie schwere Fehler in einer Entwicklungsphase gemacht werden
- Typische Fehler im Umgang mit Prognosemodellen:
    - Zeitlich jüngere Werte werden überbewertet
    - Populäre bzw. viel diskutierte Werte werden überbewertet
    - Es werden scheinbare Muster erkannt, die nicht nachweisbar sind
    - Einfluss von Wunsch- und Angst-Vorstellungen
    - Daten werden so ausgewählt oder interpretiert, wie es den eigenen Erwartungen entspricht

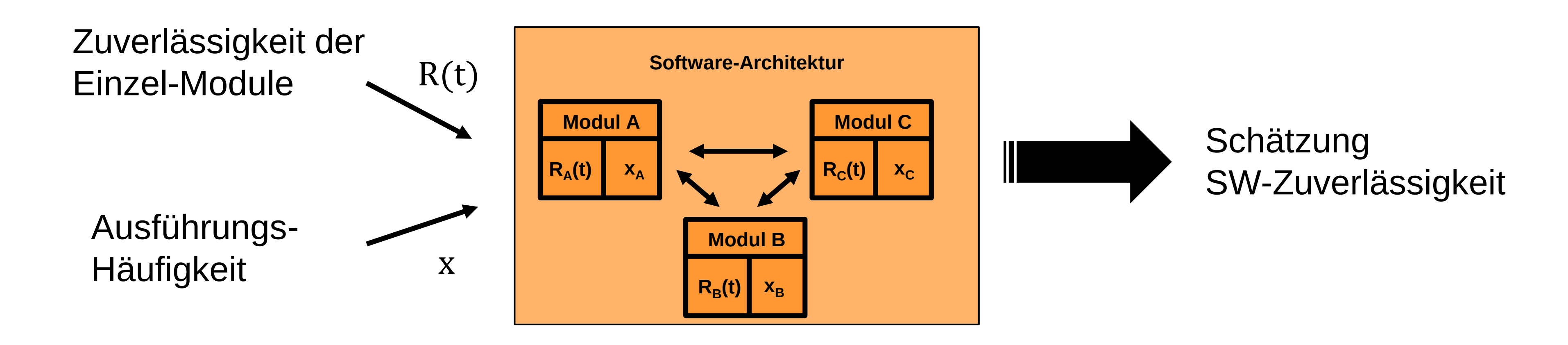
„Predicition is very difficult, especially about the future!“

(Niels Bohr (1885-1962), dänischer Physiker, Nobelpreis 1922 für die Erforschung der Atomstruktur)

Architekturbasierte Modelle - Prinzip

- Abschätzung der Zuverlässigkeit eines Entwurfs oder einer Implementierung auf Basis der jeweilig bekannten, empirischen Einzel-Zuverlässigkeiten und Ausführungs-Häufigkeiten der beinhaltenden Module

- Prinzip:

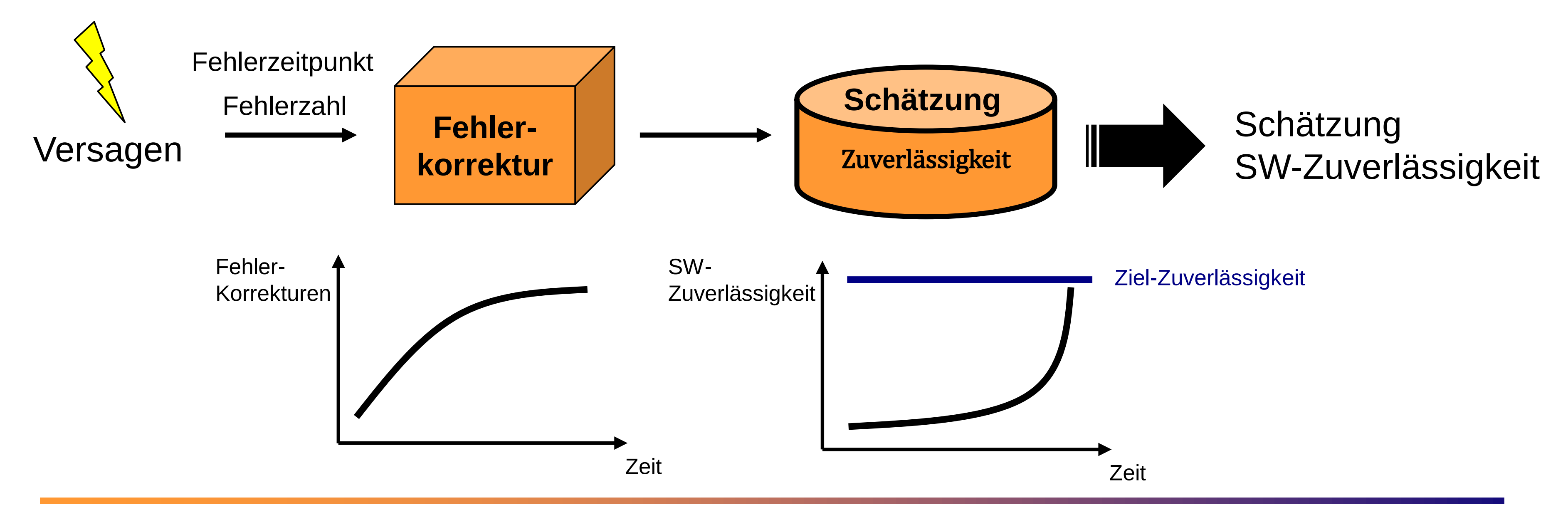


Architekturbasierte Modelle - Bewertung

- Vorteile:
  - Hohe Genauigkeit bei möglichst präzisen Eingabedaten
  - Einfluss einzelner Module auf die Gesamtheit kann analysiert werden
- Nachteile:
  - Zuverlässigkeitswerte der einzelnen Module sind häufig unbekannt
  - Bestimmung der Ausführungs-Häufigkeit ist in frühen Phasen schwierig

Zuverlässigkeitswachstumsmodelle - Prinzip

- Schrittweise Schätzung der Zuverlässigkeit auf Basis der zu Beginn (unter Annahme) getroffenen und der im Betrieb empirischen Versagenszeitpunkte von Software
- Annahme, dass nach einem Versagenszeitpunkt der Fehler korrigiert wird und bei der Korrektur keine neuen Fehler entstehen
- Prinzip:



Zuverlässigkeitswachstumsmodelle - Bewertung

- Vorteile:
  - Hohe Genauigkeit bei präziser Auswertung der empirischen Daten
    - In vorangegangen Tests
    - Im laufenden Betrieb
  - Nachvollziehbarer und verfolgbarer Entwicklungs- und Fehlerkorrekturverlauf
- Nachteile:
  - In frühen Phasen liegen nicht ausreichend empirische Daten vor
  - Es werden oftmals Annahmen getroffen, die in der Praxis nicht zutreffen (besonders in frühen Phasen)
  - Einfache bzw. direkte Übertragung der Ergebnisse von früheren System ist nicht möglich, da individuell entwickelte Software direkt vergleichbar ist

Klassifizierung nach der Zeitabhängigkeit

- Bei zeitabhängigen Modellen haben die Daten und Testergebnisse einen zeitlichen Bezug
- Prozesse der Fehlererkennung und -Korrektur zur Bestimmung der:
  - Anzahl der Fehler in einem Programm
  - Fehlerrate, d.h., der Anzahl der in einer Zeiteinheit aufgetretenen Fehler in Abhängigkeit von der Gesamtfehlerzahl
  - Mittleren Zeit zwischen zwei erkannten Fehlern, wobei die Zeit nach jeder Korrektur größer wird
- Beispiele:
  - Shooman-Modell
  - Jelinski-Moranda-Modell
  - Schick-Wolverton-Modell

Shooman-Modell - Prinzip

- Zeitabhängiges Modell zur Schätzung der:
  - Anzahl von Fehlern, die sich zu Beginn im Programm befinden
  - Anzahl der Restfehler, die sich weiterhin im Programm befinden
  - Fehlerrate und Zuverlässigkeit der Software
- Programmcode wird über einen bestimmten Zeitraum (Intervall) in Betrieb genommen und getestet
- In einem Testintervall wird eine bestimmte Fehlerzahl gefunden und korrigiert
- Es gilt dabei:

  $I \triangleq$ Anzahl der Programmanweisungen

  $t \triangleq$ Betriebszeit

  $\tau \triangleq$ Testzeit

  $H \triangleq$ fehlerfreie Testzeit

Shooman-Modell - Fehlerzahl und Zeitabhängigkeit (1/2)

- Für die Anzahl der Fehler gilt

E	$\triangleq$	(Gesamt-) Anzahl der Fehler zu Beginn
ε	$\triangleq$	Relative Fehlerzahl (Fehlerdichte), d.h.,Fehlerzahl auf I normiert
E_c , ε_c	$\triangleq$	Anzahl der korrigierten Fehler, Anzahl auf I normiert
E_r , ε_r	$\triangleq$	Anzahl der Restfehler, Anzahl auf I normiert

- Summe E der korrigierten und restlichen Fehler wird als konstant angenommen:

$$\mathbf{E = E_c(\boldsymbol{\tau}) + E_r(\boldsymbol{\tau}) = const.}$$

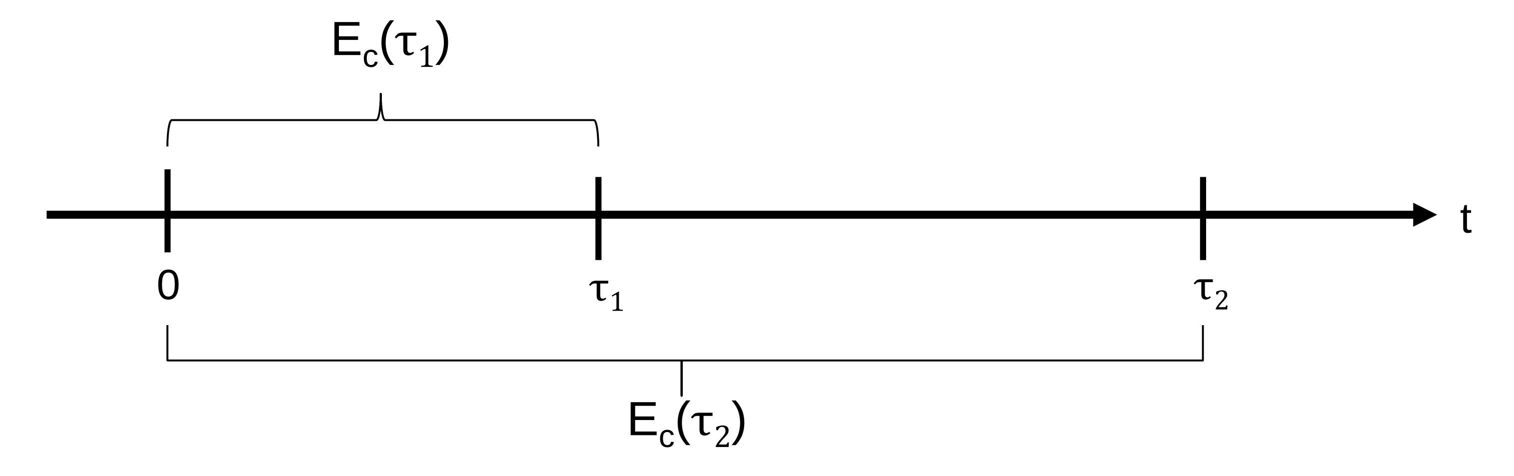
- Es gilt dabei für die auf die Anzahl der Programmanweisung bezogenen Fehler:

$$\mathbf{e_c(\boldsymbol{\tau}) = \frac{E_c(\boldsymbol{\tau})}{I}}$$

$$\mathbf{e_r(\boldsymbol{\tau}) = \frac{E_r(\boldsymbol{\tau})}{I} = \frac{E - E_c(\boldsymbol{\tau})}{I}}$$

Shooman-Modell - Fehlerzahl und Zeitabhängigkeit (2/2)

– Es werden mehrere Testintervalle betrachtet, z.B. τ_1 und τ_2:



– Es werden dabei folgende Annahmen getroffen:

- Es entstehen keine neuen Fehler bei der Korrektur
- Für die Zahl der korrigierten Fehler in den Intervallen mit $\tau_1 < \tau_2$ gilt:

$$\mathbf{E_c(\tau_1) < E_c(\tau_2)}$$

Shooman-Modell – Berechnung der Fehlerrate

– Spezifische Fehlerrate in den Test-Intervallen τ_1 bzw. τ_2:

$$\lambda_{SW1}(t) = \frac{E_c(\tau_1)}{H_1} \qquad \lambda_{SW2}(t) = \frac{E_c(\tau_2)}{H_2}$$

– Fehlerrate wird innerhalb der Intervalle bis zur nächsten Korrektur als konstant angenommen:

$$\lambda_{SW1/2}(t) \stackrel{\text{def}}{=} \lambda_{SW1/2} = \text{const.}$$

– Änderung der Fehlerrate:

$$\lambda_{SW1} - \lambda_{SW2} = \frac{c}{I}\left(E_c(\tau_2) - E_c(\tau_1)\right)$$

– Änderung der Fehlerrate pro Fehlerkorrektur:

$$\Delta\lambda_{SW} = \frac{c}{I}$$

Shooman-Modell – Proportionalitätskonstante c

- Proportionalitätskonstante c ist ein fester Wert und ergibt sich aus
  - Programmiererfahrung der Softwareentwickler
  - Dauer des Projekts
  - …
- Gegeben aus Erfahrungswerten oder wird berechnet über:
  - Proportionalität der Fehlerrate zu der Anzahl der Fehler:

$$\lambda_{SW} = c * \frac{E_r(\tau)}{I} = c * \frac{E - E_c(\tau)}{I}$$

  - Umformung führt zu:

$$c = \frac{I * \lambda_{SW}}{E - E_c(\tau)} = \frac{\lambda_{SW}}{\frac{E}{I} - \varepsilon_c(\tau)} \stackrel{\text{def}}{=} \frac{\lambda_{SW1}}{\frac{E}{I} - \varepsilon_c(\tau_1)} = \frac{\lambda_{SW2}}{\frac{E}{I} - \varepsilon_c(\tau_2)}$$

Shooman-Modell – Bestimmung der Modellparameter

– Berechnung der Gesamtzahl der im Programmcode enthaltenen Fehler:

$$\mathbf{E = I * \frac{\frac{\lambda_{SW2}}{\lambda_{SW1}} * \varepsilon_c(\tau_1) - \varepsilon_c(\tau_2)}{\frac{\lambda_{SW2}}{\lambda_{SW1}} - 1}}$$

– Berechnung der Überlebenswahrscheinlichkeit bzw. Zuverlässigkeit

$$\mathbf{R_{SW}(t) = e^{-\lambda_{SW} * t} = e^{-c * \frac{(E - E_c(\tau))}{I} * t}}$$

– Berechnung der mittleren Betriebsdauer bis zum Ausfall (MTTF)

$$\mathbf{MTTF_{SW} = \frac{1}{\lambda_{SW}} = \frac{1}{c * \frac{(E - E_c(\tau))}{I}}}$$

Shooman-Modell – Rechenbeispiel (1/4)

– Gegeben:

- I =1000 Anweisungen im Programm
- Testintervalle τ_1 = 20 Tage und τ_2 = 40 Tage
- 5 Stunden fehlerfreie Testzeit pro Tag
- Im ersten Testintervall werden 16, im zweiten 24 Fehler korrigiert

– Daraus folgt:

- Anzahl der normierten Fehler

$$\mathbf{e_c(\tau_1) = \frac{E_c(\tau_1)}{I} = \frac{16}{1000} = 0{,}016}$$

$$\mathbf{e_c(\tau_2) = \frac{E_c(\tau_2)}{I} = \frac{24}{1000} = 0{,}024}$$

Shooman-Modell – Rechenbeispiel (2/4)

- Fehlerfreie Testzeiten

$$\mathbf{H_1 = 20d * 5h = 100h}$$

$$\mathbf{H_2 = 40d * 5h = 200h}$$

- Abschätzung der Fehlerraten

$$\boldsymbol{\lambda_{SW1}(t) \stackrel{\text{def}}{=} \lambda_{SW1} = \frac{E_c(\tau_1)}{H_1} = \frac{16}{100h} = 0,16h^{-1}}$$

$$\boldsymbol{\lambda_{SW2}(t) \stackrel{\text{def}}{=} \lambda_{SW2} = \frac{E_c(\tau_2)}{H_2} = \frac{24}{200h} = 0,12h^{-1}}$$

Shooman-Modell – Rechenbeispiel (3/4)

– Bestimmung der Modellparameter:

- Gesamtzahl der Fehler

$$\mathbf{E} = \mathbf{I} * \frac{\frac{\lambda_{SW2}}{\lambda_{SW1}} * \varepsilon_c(\tau_1) - \varepsilon_c(\tau_2)}{\frac{\lambda_{SW2}}{\lambda_{SW1}} - 1}$$

$$= 1000 * \frac{\frac{0,12}{0,16} * 0,016 - 0,024}{\frac{0,12}{0,16} - 1}$$

$$= 48$$

Shooman-Modell – Rechenbeispiel (4/4)

- Proportionalitätskonstante

$$\mathbf{c = \frac{I * \lambda_{SW1}}{E - E_c(\tau)} = \frac{I * \lambda_{SW2}}{E - E_c(\tau)}}$$

$$\mathbf{= \frac{1000 * 0,16}{48 - 16} = \frac{1000 * 0,12}{48 - 24} = 5}$$

- Änderung der Fehlerrate pro Fehlerkorrektur

$$\mathbf{\Delta\lambda_{SW} = \frac{c}{I} = \frac{5}{1000} = 0,005}$$

Jelinski-Moranda-Modell

- Zeitabhängiges Zuverlässigkeitswachstumsmodell
- Zeiträume der Zuverlässigkeitsschätzung im Bereich von Monaten
- Es gelten folgende Annahmen:
  - Zu Beginn befinden sich eine Zahl von X Fehlern im Programm
  - Tritt ein Fehler auf, so wird dieser, ohne die Entstehung neuer Fehler, erfolgreich behoben
  - Fehler sind voneinander und von der Vorgeschichte unabhängig

- Benötigte Daten:
  - Anzahl der Fehler zu Beginn (Schätzung aus Erfahrungswerten)
  - Zahl der Fehler bis zu einem Ausfall
  - Zeit zwischen den Ausfällen (MTBF)

Schick-Wolverton-Modell

- Zeitabhängiges Modell zur Berechnung der Restfehlerzahl und der Fehlererkennungsrate innerhalb definierter Zeitintervalle
- Es gelten folgende Annahmen:
  - Fehlerentdeckungsrate ist proportional zu der Zahl der enthaltenen Restfehler
  - Anzahl der Restfehler nimmt exponentiell ab
  - Fehler treten zufällig auf und sind voneinander unabhängig
  - Alle Fehler sind zu jedem Zeitpunkt gleich gefährlich
  - Fehlerentdeckungsrate ist in den vorgegebenen Zeitintervallen konstant
  - Alle entdeckten Fehler werden behoben und es werden dabei keine neuen Fehler gemacht

Klassifizierung nach der Zeitunabhängigkeit

- Bei zeitunabhängigen Modellen liegt kein zeitlicher Bezug der Daten und Testergebnisse vor
- Kennzeichnend für zeitunabhängige Tests ist:
  - Große Zahl an Testdurchführungen
  - Alle Tests sind voneinander unabhängig
  - Dauer der Testdurchführung hat keine Bedeutung
- Für die Testergebnisse gilt:
  - Jeder Test ist eindeutig (in Ordnung / nicht in Ordnung)
  - Ein erkannter Fehler wird nur einmal gezählt
- Beispiele:
  - Mill-Modell
  - Lipow-Modell

Mill-Modell - Fehlersaat-Verfahren

- Zeitunabhängiges Modell zur Abschätzung der Fehlerzahl
- Vorgehen:
  - Künstliche Fehler werden in ein Programm einbaut
  - Durchführung von Programmtests mit den eingebauten Fehlern
  - Aus Testergebnisse erfolgt Ableitung der Zahl erkannter echter Fehler
  - Rückführung auf die Zahl zu Beginn im Programm enthaltener Fehler
- Annahmen:
  - Gleiche Erkennungswahrscheinlichkeit für jeden Fehler
  - Anzahl der eingebauten Fehler (Schätzung aus Erfahrungswerten) ist deutlich größer als Anzahl der echten Fehler

Mill-Modell - Rechenvorschrift

– Es gilt:

X ≙ Gesamtzahl der echten Fehler

Y ≙ Gesamtzahl der eingebauten Fehler

U ≙ Zahl der erkannten echten Fehler

V ≙ Zahl der erkannten eingebauten Fehler

– Mit dem Zusammenhang: $$\frac{X}{Y} = \frac{U}{V}$$

$$X = Y * \frac{U}{V}$$

Lipow-Modell

- Zeitunabhängiges Zuverlässigkeitswachstumsmodell
- Vorgehen:
  - Ein Testprogramm wird in N Stufen unterteilt
  - Jede Stufe besitzt eine bestimmte Anzahl von Testversuchen
  - Alle Tests einer Stufe beziehen sich auf ähnliche Elemente
  - Testergebnisse sind entweder „erfolgreich“ oder „fehlgeschlagen“
  - Ergebnisse einer Stufe werden verwendet, um das jeweilige Elemente zu verbessern
  - Kontinuierliche Verbesserung der Zuverlässigkeit

Frage zu Kapitel 5.3

Welchen Aussagen stimmen Sie zu?

☐ Basis aller SW-Zuverlässigkeitsmodelle sind empirische Daten.

☐ Architekturbasierte Modelle ermöglichen eine relativ genaue Abschätzung der Zuverlässigkeit bei Kenntnis der Eingabeparameter.

☐ Bei frühzeitigen Prognosemodellen kann die frühe Interpretation der Werte zu falschen Ergebnissen führen.

☐ Zuverlässigkeitswachstumsmodelle profitieren davon, dass Werte vorheriger Projekte direkt eingesetzt werden können.

§ 5 Softwarezuverlässigkeit

5.1 Grundlagen der Softwarezuverlässigkeit

5.2 Maßnahmen gegen Softwarefehler

5.3 Modelle der Softwarezuverlässigkeit

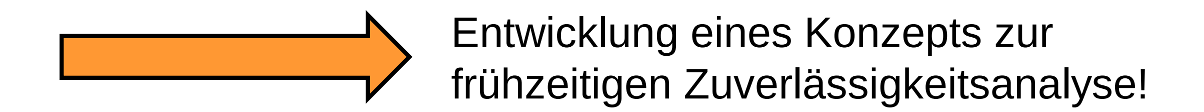
5.4 Komponentenansatz

5.5 Situationsbasierte qualitative Modellbildung und Analyse (SQMA)

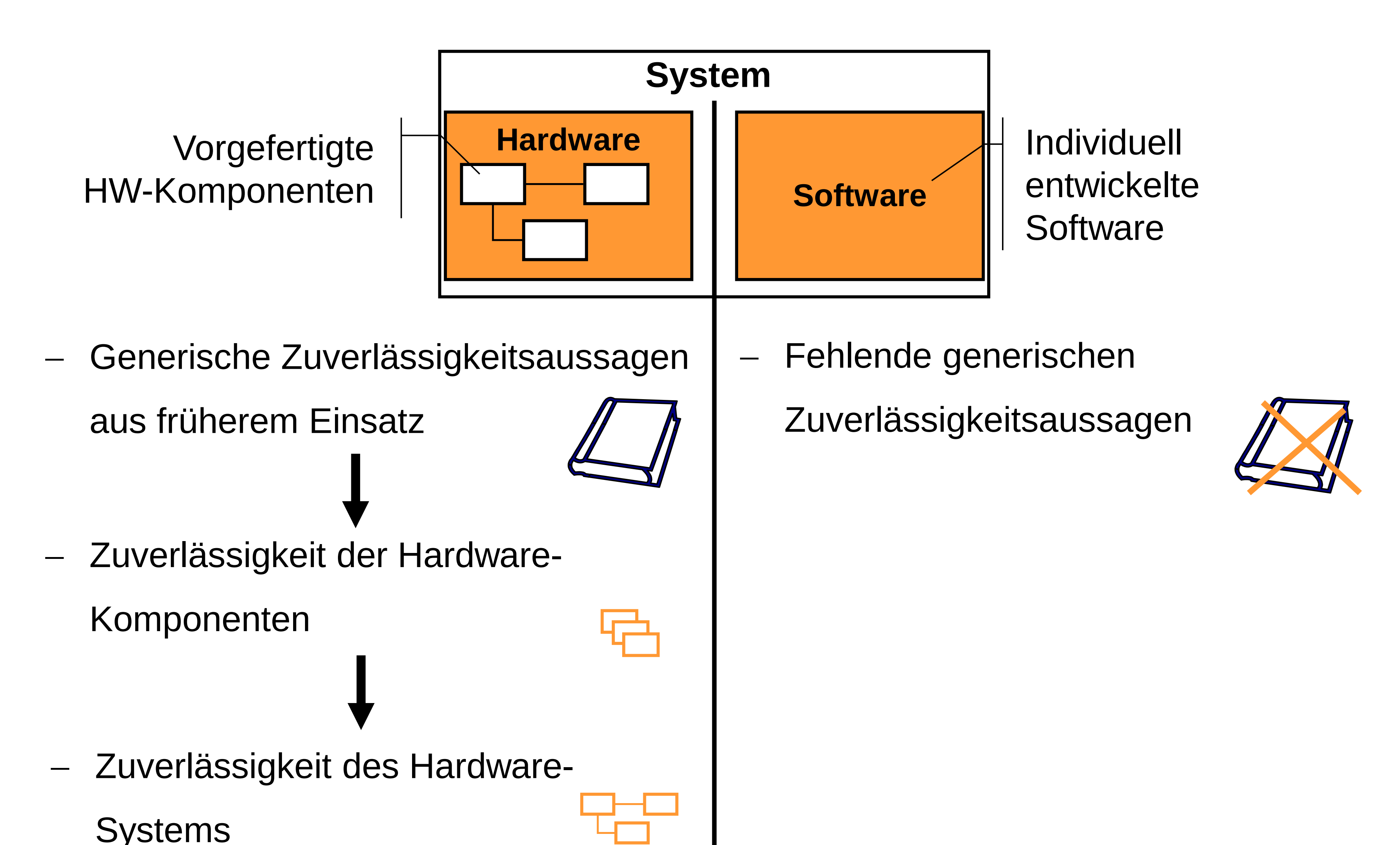
Probleme mit klassischen Zuverlässigkeitsmodelle

– Verfügbarkeit empirischer Daten
  - In frühen Entwicklungsphasen sind oftmals keine empirischen Daten über die zu entwickelnde Software verfügbar
  - Notwendige Daten werden häufig gar nicht oder nicht systematisch erfasst
  - Aufwand für den Aufbau von Datenbanken ist sehr hoch

– Anwendbarkeit der Modelle im Entwicklungsprozess
  - Keine Modelle mit guter Genauigkeit für frühe Entwicklungsphasen
  - Zuverlässigkeitsaussagen schwer interpretierbar (z.B. 10^{-3} Versagensfälle / Anforderung)

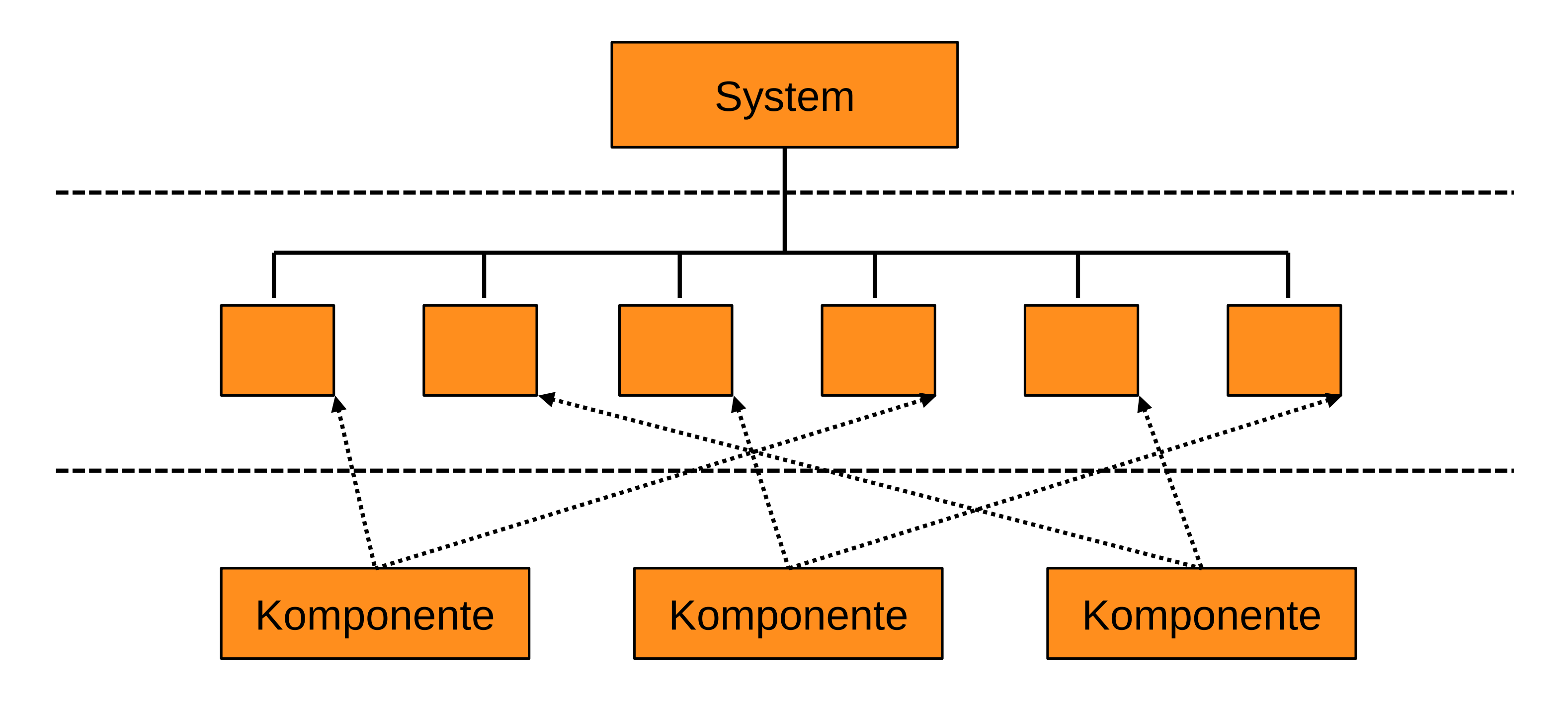


Zuverlässigkeitsanalyse von Hardware und Software

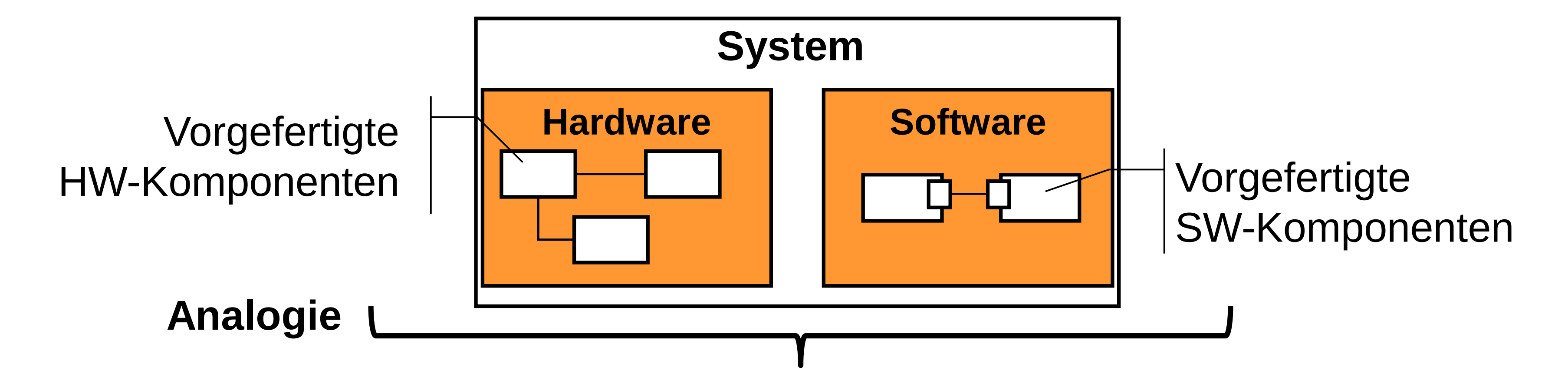


Komponentenentwicklung als Lösungsansatz

- Kleinstbausteine werden durch vorgefertigte Komponenten ersetzt, für die bereits empirische Zuverlässigkeitsdaten vorliegen
- Anwendung bisher in der Hardware-Entwicklung, nun Übertrag des Prinzips auf Software-Entwicklung



Übertragung des Komponentenansatz auf Software



- Einsatz von Komponenten in beiden Systemen
- Erfassung und Auswertung empirischer Daten!
- Ergebnis: Abhängigkeit Einflussgrößen ↔ Zuverlässigkeit

↓

- Generische Zuverlässigkeitsaussagen

↓

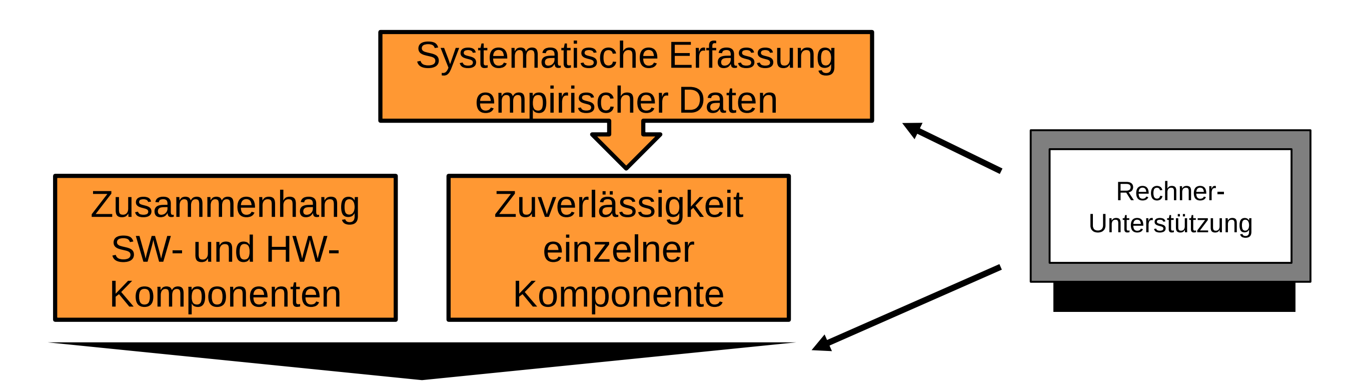
- Zuverlässigkeit des Gesamtsystems (Hardware und Software)

Erfassung und Auswertung empirischer Daten

- Ziel ist es, Erkenntnisse aus empirischen Daten früherer Test- oder Betriebszeiten auf das aktuell zu untersuchende System übertragen zu können
- Es ist aber nicht möglich Daten direkt zu übertragen
- Daher keine Entwicklung eines allgemein gültigen Modells, sondern eines Werkzeugs zur Bewertung von:
  - Produkteigenschaften unter bestimmten Randbedingungen
  - Parameter des Entwicklungsprozess



- Es gilt:

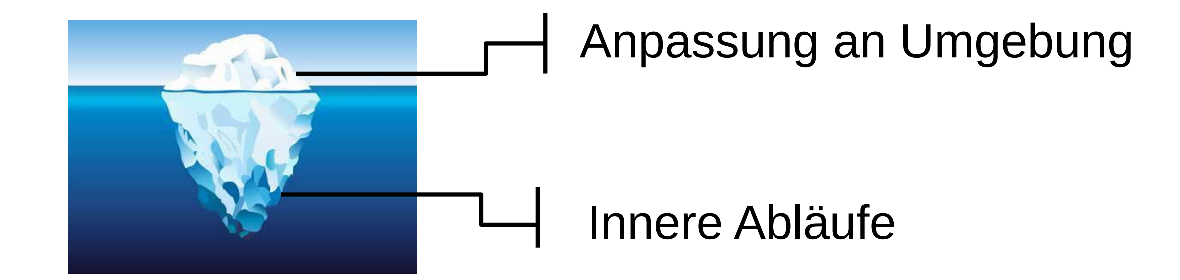


Software-System-Zuverlässigkeit

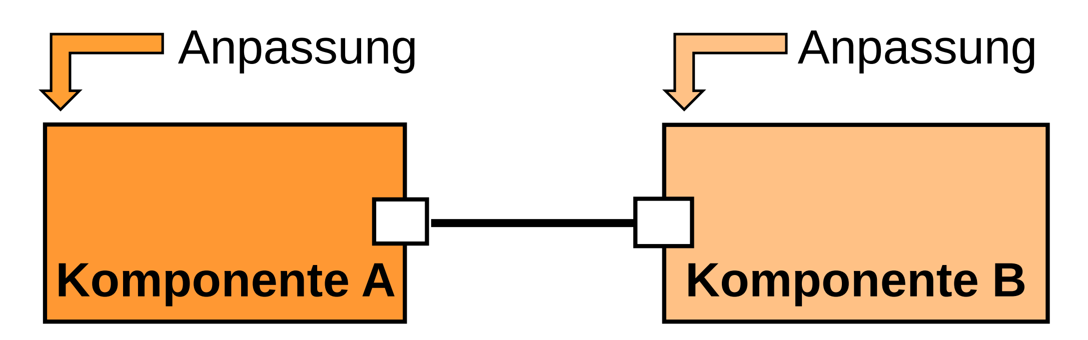
Struktur einer Softwarekomponente

- SW-Komponente ist eine unveränderliche, geschlossene Einheit, die an die jeweilige Umgebung angepasst werden muss
- Prinzip:





- Komponenten sind über Schnittstellen miteinander verbunden
- Aktivierung einer Komponente über eine andere Komponente, den Benutzer oder das Betriebssystem



Spezifische Eigenschaften von Softwarekomponenten (1/3)

- Anpassbarkeit
  - Eine SW-Komponenten muss durch geeignete Parametrierung und Konfiguration an die Aufgabe im aktuellen System angepasst werden
  - Dazu gehört auch das An- und Abschalten von Funktionen

- Funktionale Geschlossenheit
  - Funktionalität innerhalb der Komponente ist logisch zusammenhängend
  - Nach außen ist Funktionalität nur über die Schnittstellen sichtbar

- Unveränderbarkeit durch Dritte
  - SW-Komponenten werden in verschiedenen Systemen eingesetzt
  - Quellcode kann dabei vom Einsetzenden nicht verändert werden

Spezifische Eigenschaften von Softwarekomponenten (2/3)

- Verknüpfbarkeit
  - SW-Komponenten werden über Schnittstellen miteinander verknüpft, um geforderte Systemfunktionen zu erbringen
  - Klare Unterscheidung notwendig, ob:
    - Dienst erfragt oder angeboten wird
    - Ereignis gemeldet oder entgegengenommen wird
- Strukturelle Unabhängigkeit
  - SW-Komponenten können die Dienste anderer Komponenten erfordern
  - Dabei wird nur die jeweilige Funktionalität benötigt, nicht eine andere Komponenten als solche
  - SW-Komponenten sind somit strukturell voneinander abhängig

Spezifische Eigenschaften von Softwarekomponenten (3/3)

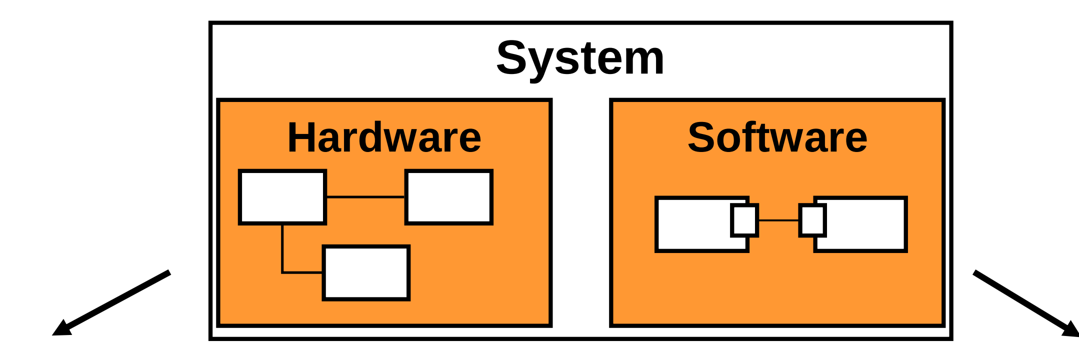
- Nebenläufigkeit
  - SW-Komponente muss in der Lage sein, parallel mit anderen Komponenten ausgeführt zu werden
- Einmaligkeit
  - SW-Komponente ist nur einmal in einem System vorhanden, kann aber mehrerer Zustände annehmen
  - D.h., sie kann eine Funktionalität in einem System in verschiedenen Zusammenhängen bereitstellen

- Offenheit
  - Spezifikation Komponenten muss gut dokumentiert und zugänglich sein
    - Funktionalität
    - Schnittstellenbeschreibung

Einsatz von Komponenten

- Wesentliche Eigenschaft einer SW-Komponenten ist die unveränderte mehrfache Verwendung
- Versagen einer Komponente ist auf Fehler zurückzuführen, die von der jeweiligen Umgebung unterschiedlich häufig aktiviert werden
- Fehler sind dabei auf Einflussfaktoren der Umgebung zurückzuführen, da die Komponente funktional geschlossen und unveränderbar ist
- Nur über die Parametrierung bzw. Anpassung an die Umgebung kann auf die Fehler Einfluss genommen werden
- Abhängigkeit zwischen Komponente und Umgebung muss bekannt sein
- Daher Beschreibung bekannter Abhängigkeiten in Komponenten-Bibliotheken

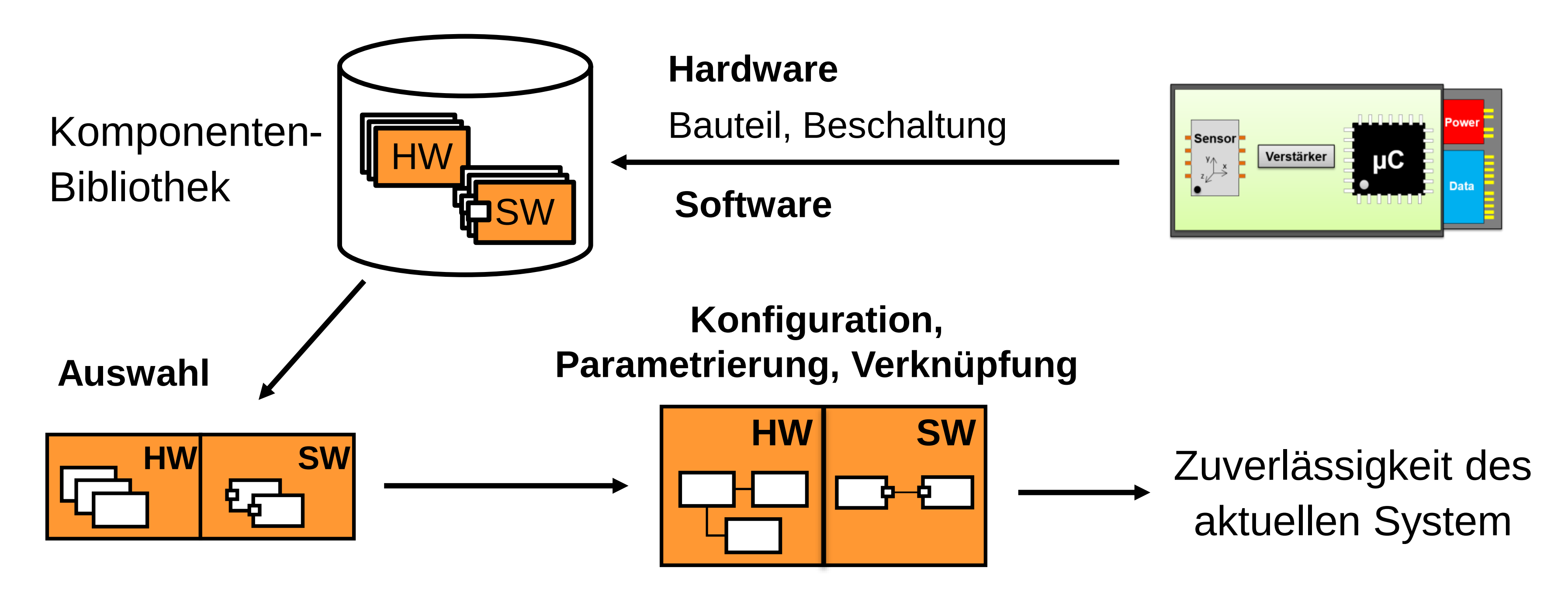
Identifikation relevanter Einflussfaktoren



- Auf Hardware-Komponenten:
  - Physikalische Umgebung
    - Temperatur,
    - Feuchtigkeit,
    - Vibration
  - Benutzung
    - Lastprofil
    - Beanspruchungen
  - Wechselwirkungen

- Auf Software-Komponenten:
  - Konfiguration bzw. Parametrierung
    - Eingaben Entwickler
    - Parameterschätzverfahren
  - Zeitliche Aktivierung
    - Externe zeitliche Aktivierung
    - Andere Komponente über Verknüpfung

Zuverlässigkeitsanalyse beim Systementwurf



Komponentenzuverlässigkeit

- Aktuelle Einflussgrößen bestimmen
- Generische Aussagen übertragen

Systemzuverlässigkeit

- Architektur analysieren
- Einzelne Zuverlässigkeiten kombinieren

Zuverlässigkeitsanalyse beim Test und Betrieb des Systems

Komponenten-Bibliothek

HW

SW

Hardware

Bauteil, Beschaltung

Software

Sensor

Verstärker

µC

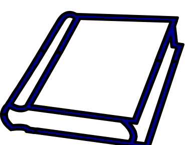
Power

Data

Mehrfacher Einsatz in verschiedenen Systemen

Empirische Daten

Generische Aussagen zur Zuverlässigkeit



Generische Zuverlässigkeitsaussagen

- Empirische Daten auswerten
- Abhängigkeit von Einflussgrößen zur Zuverlässigkeit beschreiben

Frage zu Kapitel 5.4

Welchen Aussagen stimmen Sie zu?

☐ Empirische Daten früherer Software-Systeme dürfen nicht direkt wiederverwendet werden.

☐ Eine Komponente darf nicht funktional geschlossen sein.

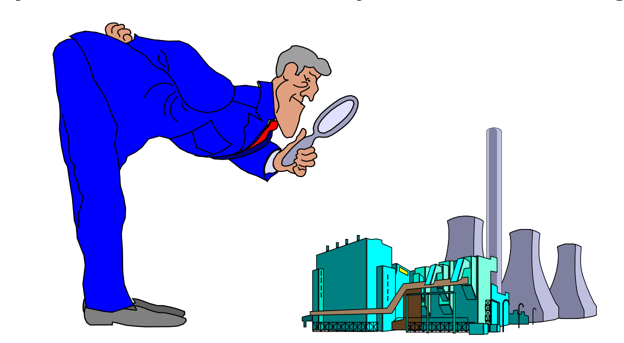
☐ Komponenten müssen immer über Parameter an ihre Umgebung angepasst werden.

☐ Eine spezifische Eigenschaft von SW-Komponenten ist es, dass diese nur einmalig im System vorhanden ist.

§ 5 Softwarezuverlässigkeit

Einführung in SQMA

"If you can not measure it, you can not manage it?"



– SQMA (**S**ituationsbasierte **Q**ualitative **M**odellbildung und **A**nalyse) ist eine am IAS entwickelte Methode zum Entwurf komplexer Systeme
  - in frühen Entwicklungsphasen.
  - trotz unvollständigen bzw. ungewissen Informationen.

– Komponentenbasierter Ansatz zur:
  - Systemmodellierung
  - Analyse des Systems

Konzept der Modellbildung mit SQMA

– Ansatz besteht aus mehreren sequentiell durchzuführenden Schritten:

1. Zerlegung des Gesamtsystems in Komponenten
2. Beschreibung der Komponenten mit qualitativen Ausdrücken
3. Modellierung des Verhaltens auf Komponentenebene
4. Komposition der Komponenten zum Gesamtsystem
5. Analyse des Gesamtsystems zur Ermittlung von Gefahren

Modellierung einzelner Komponenten – Intervall-Variablen

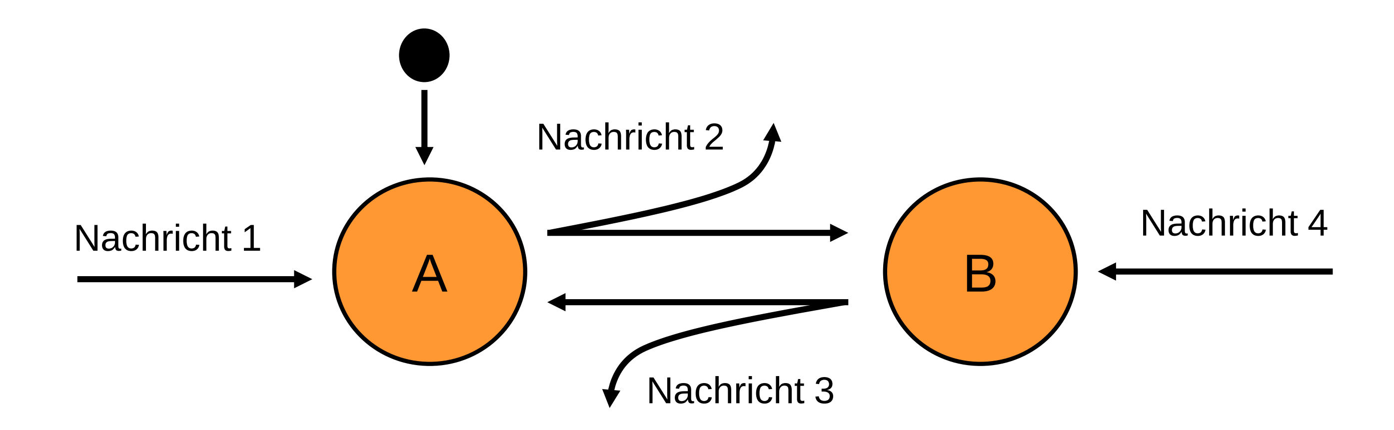
- Komponente wird aus ihrer Umgebung herausgelöst
- Es entstehen Schnittstellen zur Umgebung, an denen Größen bzw. Nachrichten auftreten:



- Nachrichten werden über qualitative Variablen beschrieben
- Variablen umfassen einen bestimmten Werteraum, der in unterschiedliche, für die Komponente typische Intervalle unterteilt wird
- Beispielsweise gilt für Nachrichten ein Intervall [X_1,X_2], wobei
  - [0,0] den Fall beschreibt, dass eine Nachricht noch nicht eingetroffen ist
  - [1,1] den Fall beschreibt, dass eine Nachricht bereits eingetroffen ist

Modellierung einzelner Komponenten - Zustandsgraph

– Verhalten der Komponente wird in einem Zustandsgraph beschrieben:
  - Tritt Nachricht 1 auf, so erfolgt direkt der Übergang in Zustand B und senden der Nachricht 2
  - Tritt Nachricht 4 auf, so erfolgt direkt der Übergang in Zustand A und Senden der Nachricht 3



Modellierung einzelner Komponenten - Situationstabelle

- Abhängigkeiten und Zusammenhänge der über Intervall-Variablen beschriebenen Größen bzw. Nachrichten werden mit Situationsregeln modelliert
- Es werden dadurch alle möglichen Kombinationen bzw. Situationen ermittelt
- Beispielhaft ein Ausschnitt aus der Situationstabelle:

Nr.	Nachricht 1	Nachricht 2	Nachricht 3	Nachricht 4	Zustand
1	[0;0]	[0;0]	[0;0]	[0;0]	A
2	[1;1]	[0;0]	[0;0]	[0;0]	B
3	[1;1]	[1;1]	[0;0]	[0;0]	B
4	[1;1]	[1;1]	[1;1]	[0;0]	A
5	[1;1]	[1;1]	[1;1]	[1;1]	A

usw.

Modellierung einzelner Komponenten - Transitionsmatrix

- Das dynamisches Verhalten der Komponente wird über Transitionsregeln beschrieben
- Es wird festgehalten, von welchen Situationen in welche Situationen übergegangen werden kann
- Es werden somit die Situationsregelübergange dargestellt werden
- Beispielhaft ein Ausschnitt aus der Transitionsmatrix:

Nr.	1	2	3	4	5
1	X	X			
2		X	X		
3			X	X	
4				X	X
5	X				X

usw.

Von den Komponenten zum System (1/2)

- Im nächsten Schritt erfolgt die Verknüpfung der modellierten Komponenten zum Gesamtsystem
- In (Netz-)listen werden dazu die Verbindungen und Kommunikationen der Komponenten untereinander festgelegt
- Verbindung entspricht sozusagen einem Nachrichtenkanal, der einen Sender mit einen Empfänger verbindet
- Es entsteht somit eine Art Ablaufdiagramm bzw. Schaltplan, wodurch die Wechselwirkungen zwischen den Komponenten beschrieben werden können
- Durch die Modellierung zugehöriger Messwerte und Stellgrößen erfolgt die Einbindung des technischen Prozess in das Ablaufdiagramm

Von den Komponenten zum System (2/2)

- Die Situationstabellen, die für die einzelnen Komponenten aufgestellt wurden, lassen sich nun reduzieren und der Situationsraum des Systems ableiten
- Der reduzierte Situationsraum enthält nun die Menge an tatsächlich möglichen Systemsituationen
- Beispielhaft zur Reduzierung auf den Situationsraum:

Nr.	Nachricht 1	Nachricht 2	Nachricht 3	Nachricht 4	Zustand
1	[0;0]	[0;0]	[0;0]	[0;0]	A
2	[1;1]	[0;0]	[0;0]	[0;0]	B
3	[1;1]	[1;1]	[0;0]	[0;0]	B
~~**4**~~	~~[1;1]~~	~~[1;1]~~	~~[1;1]~~	~~[0;0]~~	~~A~~
5	[1;1]	[1;1]	[1;1]	[1;1]	A

Nicht möglich, da Nachricht 3 nur gesendet wird, wenn Nachricht 4 angekommen ist!

usw.

Analyse und Auswertung

- Systemsituationen werden als Szenarien interpretiert, die einzelne Systemzustände beschreiben
- Systemzustände und Übergang in die Zustände können dabei möglichen Gefahrensituationen zugeordnet werden
- Bei komplexen Systemen erfolgt ein rechnerbasiertes Aufspüren von gefährlichen Systemzuständen bzw. Vorgängen
- Einheitliche Betrachtung des Systems durch Analyse des Zusammenwirkens von Software, dem technischen Prozess und dem Menschen bzw. Benutzer

Frage zu Kapitel 5.5

Welchen Aussagen stimmen Sie zu?

- ☐ SQMA erlaubt die Beschreibung komplexer Systeme trotz unvollständigen Informationen.
- ☐ Um SQMA anzuwenden, müssen ausreichend empirische Daten vorliegen.
- ☐ Vor der Modellierung über SQMA müssen die SW-Komponenten zu einem Gesamtsystem verknüpft werden.
- ☐ SQMA ermöglicht die einheitliche Analyse des Mensch-, Prozess- und Softwaremodels .

Gliederung

§ 6 Zuverlässigkeits- und Sicherheitstechnik

Risiko, Gefährdung und Gefahr…



Allgemeine Definitionen (1/2)

- Begriffe für Risiko, Gefährdung und Gefahr sind in verschiedenen ISO-, EN- und DIN-Normen definiert (z.B. DIN 31000), hier sollen nur die Erläuterungen gebracht werden
- Risiko (*risk*)
    - Die Möglichkeit, Schaden zu erleiden
    - Produkt aus Schadenshäufigkeit und Schadensausmaß
    - Risiko entspricht dem Schaden pro Zeiteinheit
    - Unterteilung:
        - Kollektives Risiko, bezogen auf Personengruppen
        - Individuelles Risiko, bezogen auf einzelne Personen
    - Sicherheit ist das Frei sein von nicht akzeptablen Risiken

Allgemeine Definitionen (2/2)

- Gefährdung (*hazard*)
  - Eine potenzielle Gefahrenquelle
  - Art der möglichen Auswirkungen einer Gefahr durch Angaben über mögliche, nicht vertretbare Personen- und/oder Sachschäden
- Gefahr (*danger*)
  - Die mögliche Schadwirkung der Gefahrenquelle oder der Zustand einer Bedrohung durch eine Gefahrenquelle
  - Zustand des Vorhandenseins von nicht vertretbaren Risiken
- Schaden (*damage [obj.], harm [pers.]*)
  - Die konkrete schädigende Auswirkung der Gefahrenquelle, als Möglichkeit oder Wirklichkeit
  - Jeder materielle oder immaterielle Nachteil, den eine Person oder Sache durch ein Ereignis erleidet

Risikokriterium Schadenshäufigkeit

– Schadenshäufigkeit ist die Anzahl von Fehlereignissen innerhalb eines Zeitraums, d.h., Anzahl der Gefahren, die im Lebenszyklus eines Produkts eintreten

Kategorie	Definition
häufig	Eine ständige Gefahr, die allgegenwärtig eintreten kann.
wahrscheinlich	Es ist zu erwarten, dass eine Gefahr oft und mehrmals eintritt.
gelegentlich	Eine Gefahr kann mehrmals eintreten.
selten	Es sinnvoll, mit dem Eintreten einer Gefahr zu rechnen.
unwahrscheinlich	Es ist anzunehmen, dass eine Gefahr nur in Ausnahmesituationen eintritt.
unvorstellbar	Es darf angenommen werden, dass keine Gefahr eintritt.

Risikokriterium Schadensausmaß

– Schadensausmaß ist ein qualitatives Maß möglicher Konsequenzen

Kategorie	Konsequenzen für eine Personen	Konsequenzen für eine Betriebs- und Dienstleistung
katastrophal	Mehrere Unfalltote und/oder zahlreiche Schwerverletzte und/oder schwere Umweltschäden.	Verlust des gesamten Systems bzw. der gesamten Funktionalität.
kritisch	Einzelne Unfalltote und/oder Schwerverletzte und/oder nennenswerte Umweltschäden.	Verlust eines bzw. mehrerer wichtiger (Teil-) Systeme.
marginal	Kleinere Verletzungen und/oder nennenswerte Bedrohung der Umwelt.	Schwere Beschädigung des Systems.
unbedeutend	Mögliche, geringfügige Verletzungen.	Geringfügige Beschädigung des Systems.

Risikostufen auf Basis der Risikokriterien

– Eine Bewertung des Risikos erfolgt vor und nach der Ausführung einer Maßnahme zur Erhöhung der Sicherheit

– Es gilt das folgende Schema

Häufigkeit	**Risikostufen**			
häufig	unerwünscht	intolerabel	intolerabel	intolerabel
wahrscheinlich	tolerabel	unerwünscht	intolerabel	intolerabel
gelegentlich	tolerabel	unerwünscht	unerwünscht	intolerabel
selten	vernachlässigbar	tolerabel	unerwünscht	unerwünscht
unwahrscheinlich	vernachlässig-bar	vernachlässigbar	tolerabel	tolerabel
unvorstellbar	vernachlässigbar	vernachlässigbar	vernachlässigbar	vernachlässigbar
	unbedeutend	marginal	kritisch	katastrophal
	Schadensausmaß			

Bewertung der Risikostufen

– Beurteilung, ob ein Risiko unter bestimmten Rahmenbedingungen akzeptabel oder nicht akzeptabel ist und ob mögliche Restrisiken vertretbar sind

Kategorie	Kriterien für anzuwendende Maßnahmen
intolerabel	Ein Risiko muss unbedingt ausgeschlossen werden.
unerwünscht	Ein Risiko darf nur akzeptiert werden, wenn eine Risikominderung praktisch nicht durchführbar ist und eine Zustimmung der für die Sicherheit zuständigen Aufsichtsbehörde vorliegt.
tolerabel	Bei geeigneter Überwachung und mit Zustimmung einer Aufsichtsbehörde kann das Risiko akzeptiert werden.
vernachlässigbar	Ohne weiter Zustimmung einer Aufsichtsbehörde darf das Risiko akzeptiert werden.

Grenzrisiko und Risikoakzeptanz

- Grenzrisiko ist das größte noch vertretbare bzw. noch akzeptable Risiko
- Grenze für das noch akzeptable Risiko wird durch gesetzliche, gesellschaftliche oder persönliche Regeln gesetzt:
  - Aktuelle Statistiken als Basis für das Grenzrisiko
  - Genormte Akzeptanzkriterien (z.B. in EN 50126)
- Das verbleibende Restrisiko ist das Risiko, das nicht durch Maßnahmen und die Wirkung aller Sicherheitsfunktionen ausgeschlossen werden kann
- Restrisiko setzt sich aus einem bekannten bzw. abschätzbaren und einem unbekannten Teil zusammen

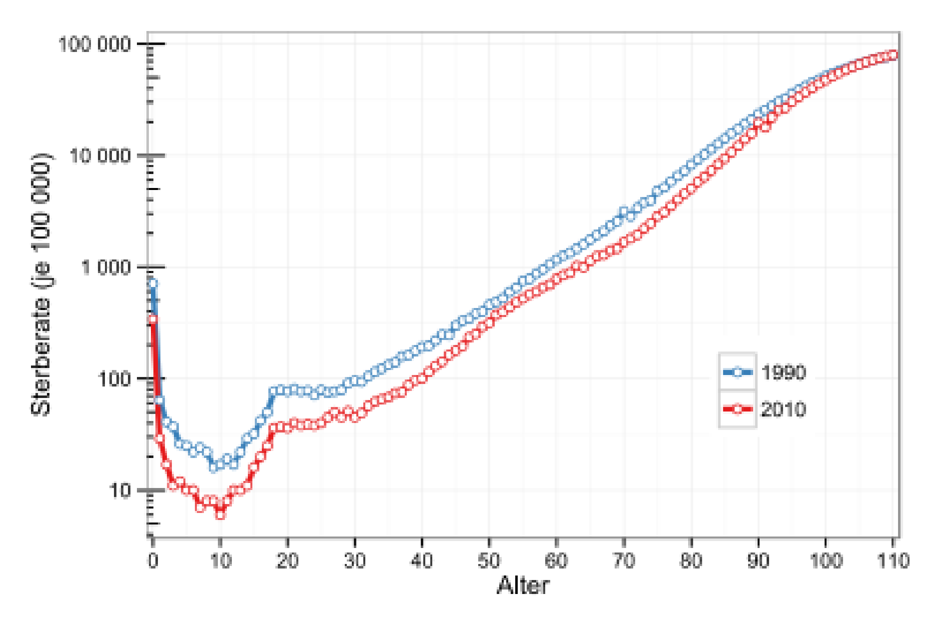
Arten der Risikoakzeptanz

- Allgemeine Risikoakzeptanz
  - Risiko, dass bei der richtigen Anwendung der nach dem Stand der Technik notwendigen Maßnahmen nicht ausschließbar ist
  - Neben der Technik an sich, sind Betriebsführung (Organisation, Personal,…) und mögliche Externe Einflüsse am Gesamtrisiko beteiligt

- Individuelle Risikoakzeptanz
  - Risiko, das man selbst bereit ist zu akzeptieren
  - Zum Beispiel Rauchen, Motorrad fahren, Sport

Mortalität

- Begriff aus der Demographie für die Sterberate, d.h., es wird die Zahl der Todesfälle bezogen auf die Anzahl der Individuen berechnet
- Darstellung der Mortalität erfolgt bezogen auf die Altersgruppen
- Beispiel Deutschland (als entwickeltes Industrieland):



Quelle: de.wikipedia.org

Lebens- und Sterberisiko

- Begriffe aus der Rechtsprechung, die mögliche Gefahren bzw. negative Ereignisse beschreiben, die zum Nachteil oder Tod von Menschen führen können
- Tod ergibt sich aus vielen verschiedenen Ursachen, wobei eine Ursachengruppe der „Technologie“ zugeordnet wird
  - Verkehr (PKW, Bahn,…)
  - Arbeitsmaschinen
  - Heimwerken
  - Unterhaltung und Sport
- Jede Ursachengruppe hat einen bestimmten Prozentsatz von Toten/Jahr
- Entsprechend der Altersverteilung der Mortalität wird das Risiko bzw. der jeweilige Prozentsatz als „Endogene Sterblichkeit/Mortalität R“ bezeichnet

Minimale Endogene Mortalität (MEM)

- MEM ist das Maß für das unvermeidliche und akzeptierte Risiko, dass Personen durch den Einsatz einer Technologie zu Tode kommen
- Wird in der Norm EN 50126 konkret beschrieben und orientiert sich an der Mortalität der Gruppe der 5- 15jährigen in einem entwickelten Industrieland:

$$R_m = 2 \cdot 10^{-4} \text{ Todesfälle/(Person x Jahr)}$$

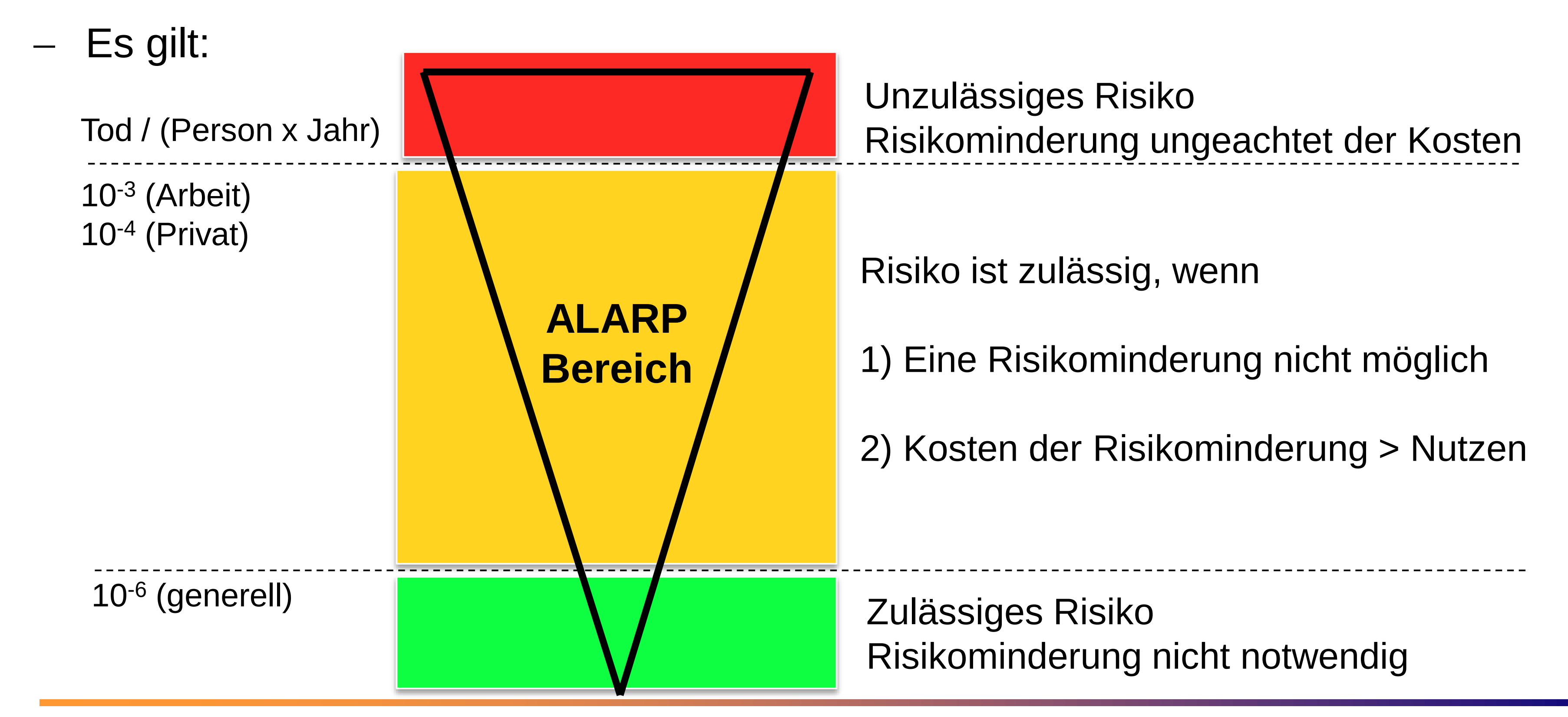
- Es wird daraus die Regel abgeleitet, dass Gefahren, die sich durch ein neues technologisches System (z.B.: Verkehr) ergeben, zu keiner nennenswerte Erhöhung der minimalen endogenen Mortalität führen
- In der Praxis akzeptierte Werte sind dabei:
  - $R_x < 10^{-5}$ Todesfälle/(Person x Jahr)
  - $R_x < 10^{-4}$ Schwerverletzte/(Person x Jahr)
  - $R_x < 10^{-3}$ Leichtverletzte/(Person x Jahr)

Bewertung des Risikoakzeptanzkriterium MEM

- Vorteile des Ansatz:
  - Garantiert feste Grenzwerte, unabhängig vom Nutzungsverhalten sowie den individuellen Risiken eines Nutzers
  - Ermöglicht den Vergleich verschiedener (Teil-)Systeme
  - Erleichtert die Arbeit von Sicherheitsaufsichtsbehörden, indem Entscheidungskriterien zur Freigabe eines Systems gegeben sind
  - Festes Maß, das sich nicht an gesellschaftliche oder politische Gegebenheiten anpassen lässt (Sensibilität aufgrund aktueller Unfälle)
- Nachteil des Ansatz:
  - Verallgemeinerung, da das Systemverhalten im Betrieb nicht genauer betrachtet wird (Nutzungsdauer, Bedienpersonal,…)

ALARP-Prinzip („As Low As Reasonably Practicable")

- Prinzip der Risikoreduzierung, nach dem ein Risiko durch ein zusätzliches System so niedrig wie vernünftigerweise praktikabel sein soll
- System muss unter dem möglichen Grad der Wirtschaftlichkeit sicher sein, die Kosten für eine Risikominderung müssen auf wirtschaftlichen Basis beruhen
- Es gilt:



GAMAB („Globalement Au Moins Aussi Bon“)

- Prinzip, das ein neues System mindestens so sicher bzw. risikoarm sein muss, wie ein bereits existierendes, vergleichbares System
- Mindestanforderung an ein neues System ist der derzeitige Sicherheitsstandard
- Technischer Fortschritt baut auf den Mindestanforderungen auf
- Vorgehen:
  - Ableitung statistischer Daten aus bisher eingesetzten Systemen
  - Charakterisierung des neuen Systems durch eine Reihe von Parametern
  - Untersuchung der System-Parameter mit Hilfe der statistischen Daten der bisher eingesetzten Systeme
  - Geschätzten Daten des neuen Systems entsprechen oder übertreffen Daten der bisherigen Systeme

Frage zu Kapitel 6.1

Welchen Aussagen stimmen Sie zu?

☐ Ist ein Risiko tolerabel, so kann es bei geeigneter Überwachung und mit Zustimmung einer Aufsichtsbehörde akzeptiert werden.

☐ Das individuelle Risiko ist das Risiko, das man selbst bereit ist zu akzeptieren.

☐ Nach dem GAMAB Prinzip muss ein System so risikoarm sein, dass es gerade noch praktikabel ist.

☐ Ein System orientiert sich bei der Risikobewertung nach dem ALARP-Prinzip an vergleichbaren Systemen.

§ 6 Zuverlässigkeits- und Sicherheitstechnik

Methoden und Techniken

- Erhöhung der Sicherheit um Gefährdungen auszuschließen
  - Fehlerausschluss
  - Fehlersicher (*fail-safe*)
  - Fehlererkennung und Überführung zur sicheren Seite
- Erhöhung der Zuverlässigkeit um die Ausfallwahrscheinlichkeit zu reduzieren
  - Bessere Komponenten
  - Besser Systemstruktur
  - Redundanzstrukturen
- Zuverlässigkeits- und Sicherheitstechnik ist die Kombination beider Methoden und Techniken, wobei gilt:

Echte-Fail-Safe-Verfahren

- Konstruktiv festgelegte Ausfallrichtung durch die Ausnutzung von physikalischen Effekten
- Sichere Ausfallrichtung durch Zusammenschaltung von Komponenten
- Keine Ausfall- oder Fehlererkennung wird verwendet
- Beispiel: Ruhestromprinzip bei Relaisschaltungen
  - System wird auch in Ruhe mit ständigem, definierten Strom betrieben
  - Aktion wird veranlasst, wenn Strom außerhalb eines Toleranzbereich ist oder eine Unterbrechung des Stromfluss vorliegt

- Bei Software-Systemen nicht realisierbar, da auch unbekannte Fehler sicherheitsgerichtet wirken müssten

Quasi-Fail-Safe-Verfahren

- Bei Erkennung eines Fehlers erfolgt die Einleitung von sicherheitsgerichteten Maßnahmen (unterbrechungsfreie Stromversorgung)
- Verwendung einer Ausfall- oder Fehlererkennung
- Beispiel: Zweikanalige Signalverarbeitung
  - Versorgung und Kommunikation über parallele Einheiten
  - Systemfunktionalität gegeben trotz Ausfall einer Einheit

- Bei Software-Systemen erfolgt Ausfall- oder Fehlererkennung über:
  - Programmablaufüberwachung
  - Rechenzeit- bzw. Laufzeitüberwachung
  - Plausibilitätsprüfungen
- Enge Verbindung bzw. Zusammenwirken von Hardware und Software

Allgemeine Beispiele von Strategien

- Hardware-Strategien
    - Verbesserung von Material, Konstruktion und Herstellungsprozess
    - Auswahl besserer Komponenten
    - Redundanz
    - Schutz gegen Umwelteinflüsse

- Software-Strategien
    - Geeignete Methoden der Softwareentwicklung
    - Geeignete Tools zur Softwareentwicklung
    - Wiederverwendung von Software
    - Selbstüberwachung

Perfektionsstrategie und Non-Perfektionsstrategie

– Strategien der Zuverlässigkeits- und Sicherheitstechnik lassen sich grundsätzlich in zwei Klassen unterteilen:

Perfektionsstrategie	**Non-Perfektionsstrategie**
– Fehlervermeidung –	– Fehlertoleranz –
Ausschließen von Fehlern **vor der Inbetriebnahme** eines Systems	System kann die spezifizierte Funktion **nach der Inbetriebnahme** trotz vorhandener Fehler erfüllen

Zeitpunkt der Inbetriebnahme

Perfektionsstrategie (1/2)

- Der Einsatz von Perfektionsstrategien richtet sich nach der Wirtschaftlichkeit eines Produkts und ist abhängig von geltenden Normen bzw. Vorschriften
- Ziel ist das generelle Ausschließen von Fehlern über:
  - Maßnahmen gegen Ausfälle aufgrund von bestimmten Ausfallmechanismen
  - Maßnahmen gegen Ausfälle aufgrund von Störeinflüssen

Perfektionsstrategie (2/2)

– Nach der Perfektionsstrategie gilt für den Entwicklungsprozess die Einhaltung bzw. Durchführung von:
  - Konstruktiven Maßnahmen
  - Sorgfältiger Entwurf
  - Erhöhte Anzahl von Tests
  - Ausgereifte Verifikations- und Testmethoden

Non-Perfektionsstrategie (1/2)

- Verhinderung gefährlicher Auswirkungen von Fehlern und Ausfällen, die im laufenden Betrieb auftreten können, durch Diagnose-, Erkennungs- und Redundanzmaßnahmen
- Non-Perfektionsstrategie erfolgt immer nur ergänzend zur Perfektionsstrategie
- Grund dafür ist:
  - Menschliche Bedienhandlungen sind im Fehler- bzw. Störungsfall immer als kritisch anzusehen
  - Reduzierung der Wahrscheinlichkeit von Mehrfachfehlern, die auch für die Toleranzstrategien kritisch sind

Non-Perfektionsstrategie (2/2)

– Maßnahmen der Non-Perfektionsstrategie werden eingesetzt zur:

- Fehlerausgrenzung, d.h., eine fehlerhafte Komponente wird durch eine fehlerfreie / funktionsfähige Komponente ersetzt.
- Fehlerbehebung, d.h., Rücksetzung auf einen gesicherten und bekannten vorherigen, fehlerfreien Zustand
- Fehlermaskierung, d.h., Aufbau redundanter Strukturen, sodass ein Mehrheitsentscheider definiert, welches Ereignis korrekt ist.
- Fehlerdetektion und -korrektur

Anwendungs-Beispiel: Ausfall eines Bauteils

- Betrachtung physikalischer Fehler, die zu dem Ausfall des Bauteils führen
- Maßnahmen der Perfektionsstrategie:
  - Maßnahmen gegen Ausfälle aufgrund von Ausfallmechanismen
    - Überdimensionierung des Bauteils
    - Verwendung eines hochwertigen, vorgealterten Bauteils
  - Maßnahmen gegen Ausfälle aufgrund von Störeinflüssen
    - Abschirmung
    - Räumliche Trennung von Energie- und Signalleitungen
- Maßnahmen der Non-Perfektionsstrategie:
  - Redundante Struktur von Sensoren innerhalb einer Sensoreinheit

Frage zu Kapitel 6.2

Welchen Aussagen stimmen Sie zu?

☐ Detaillierte Prüfung des Systems ist Teil der Perfektionsstrategie.

☐ Redundanz ist Teil der Perfektionsstrategie.

☐ Potentialtrennung ist Teil der Perfektionsstrategie.

☐ Die Verwendung mechanischer Konstruktionen mit einer hohen Steifigkeit ist Teil der Perfektionsstrategie.

§ 6 Zuverlässigkeits- und Sicherheitstechnik

Risikoanalyse

– Risikoanalyse (*risk assessment*) ist ein Mittel zur Bewertung von Situationen, Vorhaben, Systeme oder Ereignissen in allen möglichen Bereichen des Lebens

– Primäres Ziel in der Sicherheits- und Zuverlässigkeitstechnik ist die Vermeidung negativer Ereignisse durch Präventionsmaßnahmen, die aus Methoden bzw. Verfahren der Risikobewertung abgeleitet werden

– Anwendungsgebiete von Risikoanalysen sind beispielsweise:
  - In der Entwicklung zur Identifikation von Risiken neuer Systeme
  - In der Politik zu Erhebung von Risikostatistiken
  - In Banken zur Bestimmung von risikobehafteten Kundensegmenten

– Hier betrachtete Risikoanalyse befassen sich mit der Sicherheit von Systeme:
  - PSA/PRA
  - PAAG

Einführung in PSA/PRA (1/2)

- Probabilistische Sicherheitsanalyse (*Probability Safety Assessment* ‚PSA), auch probabilistische Risikoanalyse (*Probability Risk Assessment,* PRA), ist ein quantitatives Verfahren zur Risiko-Untersuchung mittels probabilistischer bzw. wahrscheinlichkeitsbasierender Methoden
- Entwickelt im Rahmen der amerikanischen Reaktorsicherheitsstudie „WASH 1400“ im Jahr 1975
- Methodischen Ansätze der Studie wurden u.a. in Deutschland übernommen, überarbeitet und gelten als Referenzanalyse für Kernkraftwerke:
    - Stufe 1: Ermittlung der Häufigkeit von Kernschäden im Reaktor
    - Stufe 2: Ermittlung der Abläufe im Falle einer Kernschmelze
    - Stufe 3: Analysen möglicher Folgen außerhalb des Kraftwerks

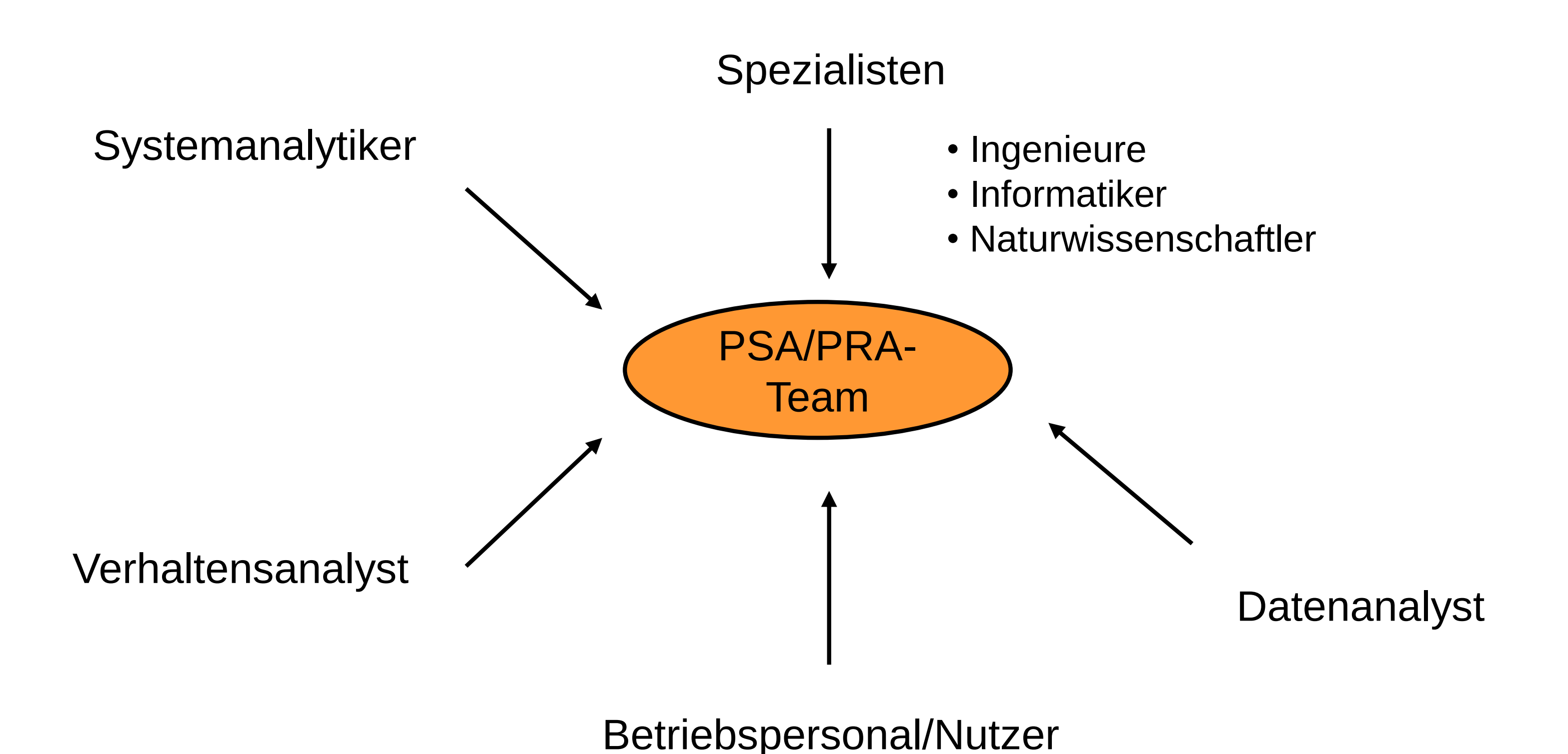
Einführung in PSA/PRA (2/2)

- PSA/PRA bedient sich bekannter Methoden (FTA, FMEA,…) und wird heute in allen Industriebereichen eingesetzt (Luftfahrt, Bahn, Chemie-Industrie)
- Wesentliche Kernfragen der PSA/PRA sind:
  - Was kann versagen?
  - Wie wahrscheinlich ist es?
  - Was sind die Auswirkungen?
- Probabilistische Eingangsgrößen der Methode werden aus Betriebserfahrung (z.B. einer Anlage) oder von vergleichbaren Systemen gewonnen
- Beispiele für Eingangsgrößen sind:
  - Häufigkeit der störfallauslösenden Ereignisse
  - Ausfallraten der Komponenten
  - Verfügbarkeit bzw. Nichtverfügbarkeit der Teilsysteme
  - Fehlerraten (Personenhandlungen, Redundanzen,…)

Ablauf von PSA/PRA (1/2)

– Tiefe des Informationsgehalts der PSA/PRA ist abhängig vom jeweiligen Anwendungsgebiet und der beteiligten Personen

– Idealerweise besteht ein PSA/PRA-Team aus:



Ablauf von PSA/PRA (2/2)

Herausforderung „menschliche Handlungen“

– Grundlegende Problematik quantitativer Methoden ist der Umgang bzw. die Bewertung menschlicher Handlungsweisen
– Menschliche Handlungen finden vor, während und nach einem Ereignis statt:
  - Als auslösendes Ereignis bzw. Ursache
  - Beinflussend mit Folge einer Eindämmung oder Verschlimmerung
  - Als Reaktion auf ein Ereignis
– Handlungen müssen ebenso über Wahrscheinlichkeiten abgeschätzt werden
– Basis sind statistische Daten, die aber häufig fehlen oder mangelhaft sind
– Daher Verfahren, die eine Kombination beinhalten aus:
  - Statistiken
  - Extrapolation verschiedenster Informationsquellen
  - Expertenbeurteilen (Soziologen, Psychologen,…)

Einführung in PAAG (1/2)

- Verfahren zur qualitativen Untersuchung der Sicherheit eines Systems
- Normung nach IEC 61882 als HAZOP-Verfahren (*Hazard and Operability Study*)
- Es gilt das Vorgehen nach dem Prinzip:
  - **P**rognose eines Ereignisses, das auftreten könnte
  - **A**uffinden der Ursache
  - **A**bschätzen der Auswirkungen
  - **G**egenmaßnahmen ableiten
- Entwickelt in der Chemieindustrie in den 1970er Jahren in England
- Berufsgenossenschaft Chemie setzte das Verfahren für Deutschland um
- Seit den 1990er wird das Verfahren auch in der Softwareentwicklung eingesetzt

Einführung in PAAG (2/2)

- PAAG kann durchgeführt werden, wenn ein erster Entwurf eines System vorliegt
- Verfahren begleitet den gesamten Entwicklungsprozess
- Experten-Team aus erfahrenen Mitarbeitern (Bereiche Entwicklung, Betrieb, Wartung und Pflege) definieren das Soll-Verhalten des Systems
- Auf Basis des Soll-Verhalten werden Parameter verändert, sodass Abweichungen vom normalen Prozessbetrieb auftreten, die Gefährdungen darstellen können
- Moderator führt das Experten-Team systematisch durch das Verfahren, indem Leitwörter mit den veränderlichen Prozess-Parameter verbunden werden, welche den Fokus der Analyse auf die Ursachen von Gefahren richtet

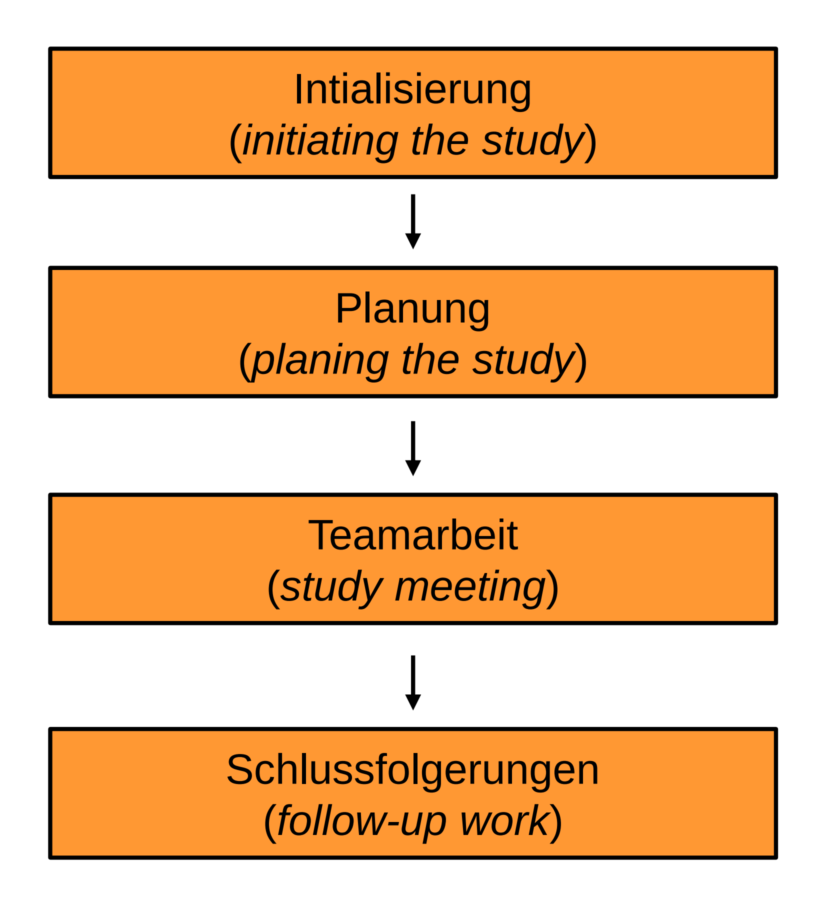
PAAG Leitwörter

– Definierte Leitwörter (*guidewords*) sind dabei:

Leitwort (de)	Guideword (eng)	Erklärung
Nein, nicht	No	Abweichung vom Soll-Verhalten
Mehr	More	Quantitativer Zuwachs
Weniger	Less	Quantitative Abnahme
Sowohl..als auch…	As well as	Zusätzliche Ereignisse zum Soll-Verhalten
Teilweise	Part of	Soll-Verhalten erfolgt unvollständig
Umkehrung	Reverse	Gegenteiliges als das Soll-Verhalten
Anders als	Other than	Etwas anders als das Soll-Verhalten
Früher/später	Earlier/later	Ein nicht erwarteter Zeitpunkt einer Aktion
Zuvor/danach	Before/after	Eine nicht erwartete Einordnung in eine Sequenz
Schneller/langsamer	Faster/slower	Eine nicht erwartete Änderung der Ablauf- bzw. Ausführungsgeschwindigkeit

Durchführung von PAAG

– Unterteilung in vier Phasen (nach Felix Redmill, Buch „*System Safety: HAZOP and Software-HAZOP*“, 1999) :



Initialisierung und Planung

– PAAG Verfahren wird von einem Verantwortlichen (*initiator*) gestartet, in dem dieser einen Leiter (*leader*) benennt

– Aufgaben des Leiters sind:

- Auswahl des Teams (Experten, Nutzer und Protokollant)
- Auswahl der Leitwörter
- Planung der Teamarbeit
- Moderation der Diskussionsrunden

Teamarbeit und Schlussfolgerung

- Moderator leitet die Diskussion
- Es wird dabei eine Tabelle angelegt, die Abweichungen vom Sollverhalten enthält:

System	Komponente	Parameter	Leitwort	Ursache	Folge	Maßnahme
µC-Einheit	Mikrocontroller	Spannungs-Versorgung	NO	Verschleiß	Ausfall	Redundanz

- Teammitglieder schlagen im Anschluss an die Diskussion mögliche Maßnahmen vor, wie kritische Fehlerfälle vermieden werden können
- Bis zur nächsten Diskussionsrunde werden die Maßnahmen umgesetzt
- Es erfolgt eine erneute Diskussion und somit die Probe, ob die kritischen Fehlerfälle neutralisiert wurden
- Gesamter Vorgang wird dokumentiert:

Frage zu Kapitel 6.3

Welchen Aussagen stimmen Sie zu?

- ☐ Ziel von Risikoanalysen ist die Ableitung präventiver Maßnahmen.
- ☐ PSA/PRA und PAAG sind beides qualitative Modelle der Risikoanalyse.
- ☐ Erster Schritt einer PSA/PRA ist die Erstellung eines Modells.
- ☐ Bei PAAG erfolgt eine Bewertung von Risikosituationen über Parameter.

§ 6 Zuverlässigkeits- und Sicherheitstechnik

6.1 Risiko, Gefährdung und Gefahr

6.2 Strategien der Zuverlässigkeits- und Sicherheitstechnik

6.3 Sicherheits- und Risikoanalysen

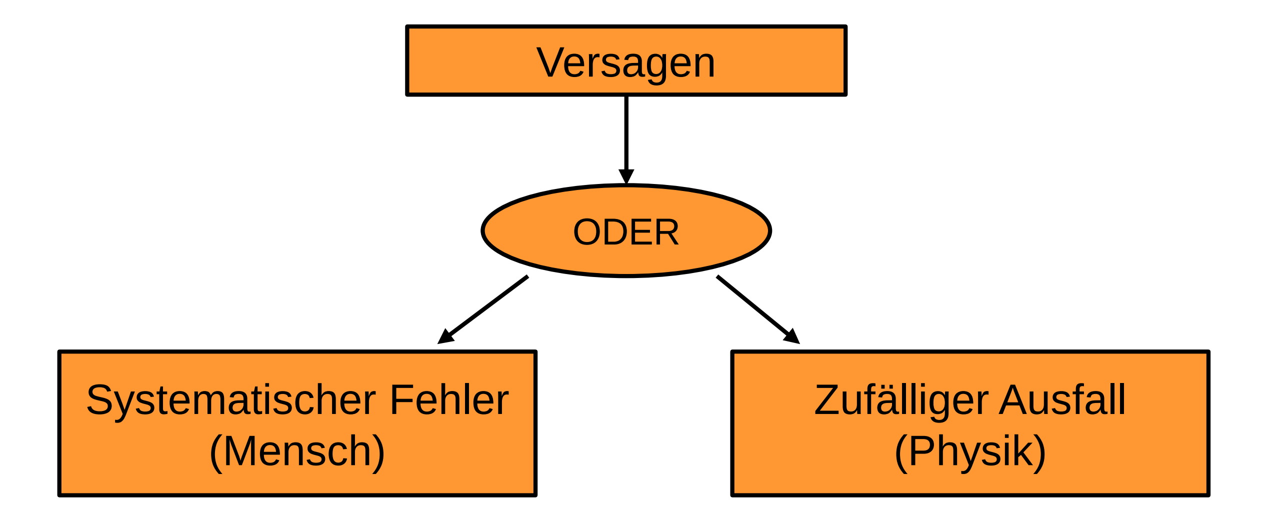
6.4 Ermittlung des Safety Integrity Levels (SIL)

Definition eines Sicherheitsgrads

– Ein Sicherheitsgrad ist eine Einordnung dafür, wie ein sicherheitsrelevantes System eine Sicherheitsanforderung erfüllt

– Sicherheitsgrade können in zwei Gruppen unterteilt werden

 - Hinsichtlich systematischer Fehler
 - Hinsichtlich zufälliger Ausfälle



Safety Integrity Levels (1/2)

- Safety Integrity Levels (SIL) sind vier diskrete Stufen von Sicherheits-Anforderungen bzw. -Integritäten an eine Sicherheitsfunktion
- Normung nach DIN 61508 (Sicherheitsgrundnorm)
- Bestimmung der jeweiligen SIL-Stufe über die Risikoanalyse eines Ausfalls des technischen Systems und der davon ausgehenden Ereignisse
- Für jede Stufe ergeben sich sicherheitsgerichtete Entwicklungsprinzipien die eingehalten werden müssen, um das Risiko einer Fehlfunktion zu vermindern
- Nach Bestimmung einer SIL-Stufe können, falls nötig bzw. möglich, Maßnahmen abgeleitet und durchgeführt werden, sodass die SIL-Stufe reduziert werden kann

Safety Integrity Levels (2/2)

- Das Vorgehen der SIL-Einstufung ist nicht über ein absolutes Verfahren vorgegeben, sondern erfolgt anhand der qualitativen Abschätzung von Risikoparametern:
  - Schadensausmaß
  - Häufigkeit der Gefahr bzw. des Aufenthalts von Personen im Gefahrenbereich (Aufenthaltsdauer)
  - Möglichkeit, die Gefahr abzuwehren oder den Schaden zu begrenzen
  - Eintrittswahrscheinlichkeit eines gefährlichen Ereignisses
- Beurteilung ist schwer, da Umgang mit einem Ereignis oder Einordnung eines Umstands oft nicht eindeutig klassifizierbar
- Einfaches, aber treffendes Beispiel: Anfahren „am Berg“
  - Unterschiedliche Auffassung von „Berg“, wenn eine Person aus Küsten- oder Alpen-Region stammt

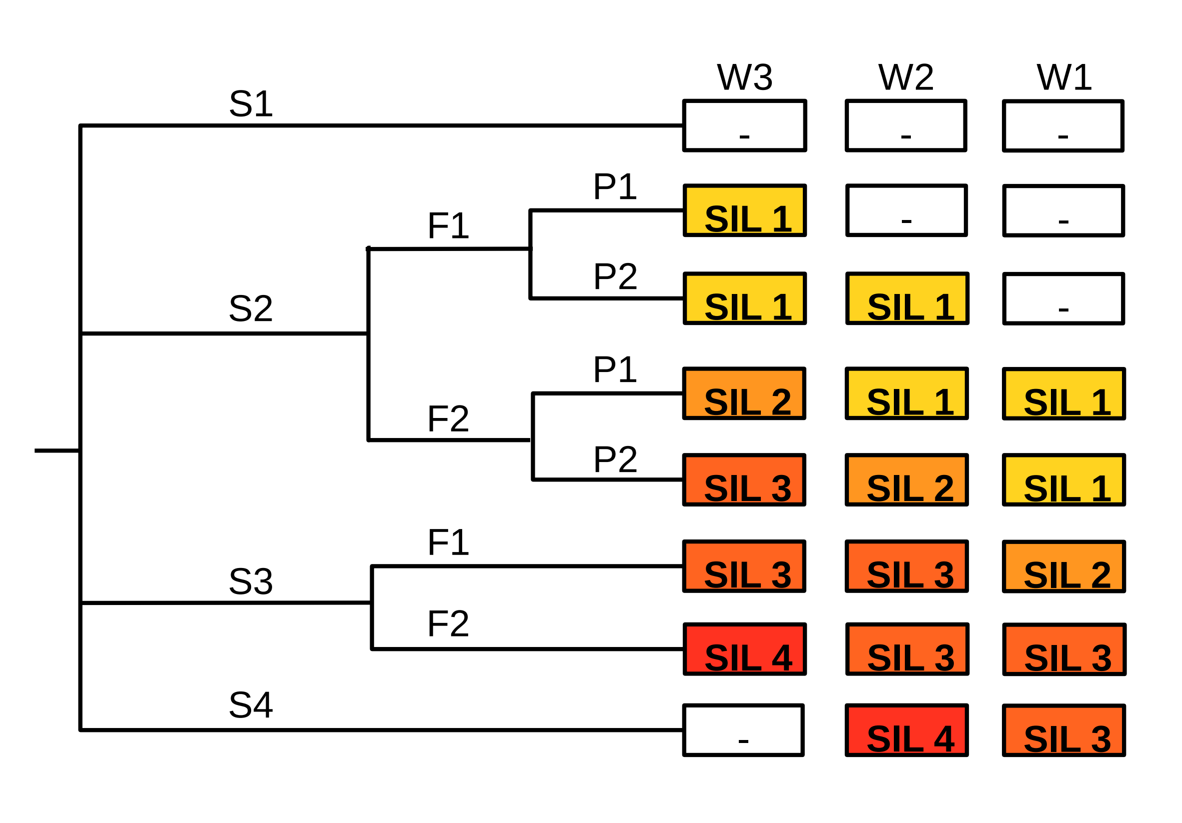
Ermittlung des Safety Integrity Levels

Werte	Bedeutung
S1	leichte Verletzungen
S2	Tod einer Person / ernste dauerhafte Verletzung
S3	Tod mehrerer Personen
S4	Tod sehr vieler Personen

F1	selten bis öfters und/oder kurze Aufenthaltsdauer
F2	häufige und/oder lange Aufenthaltsdauer

P1	möglich unter bestimmten Bedingungen
P2	kaum möglich

W1	sehr unwahrscheinlich
W2	unwahrscheinlich
W3	wahrscheinlich

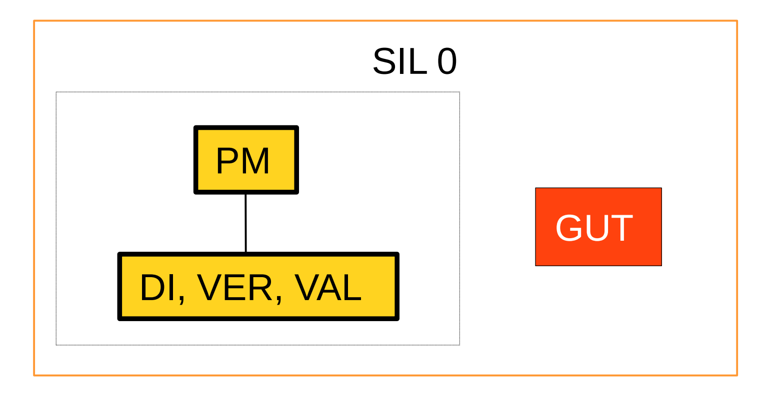


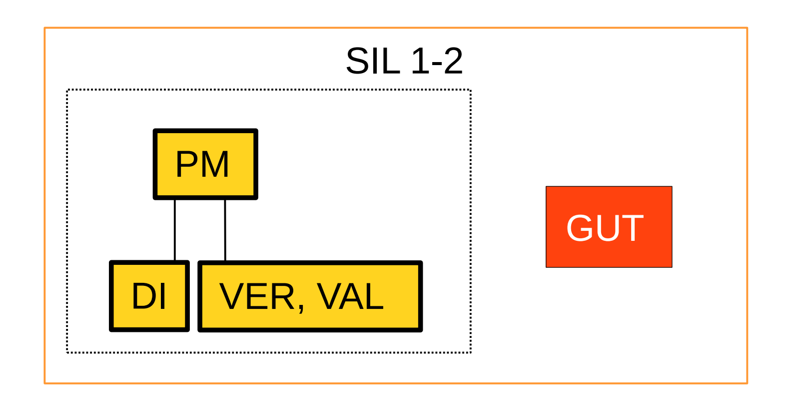
Unabhängigkeit nach SIL

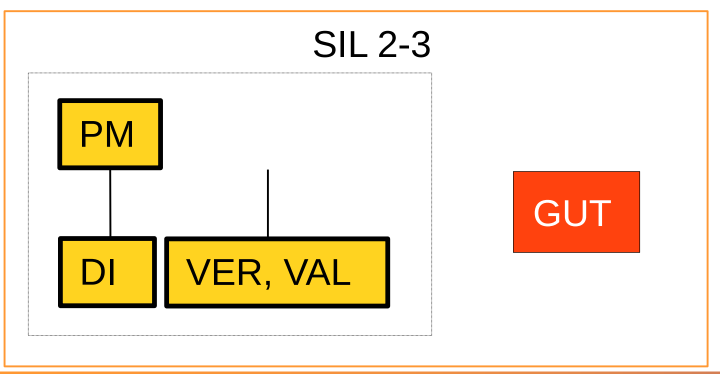
PM	≙	Projektmanager
DI	≙	Designer/Implementierer
VER	≙	Verifizierer
VAL	≙	Validierer
GUT	≙	Gutachter

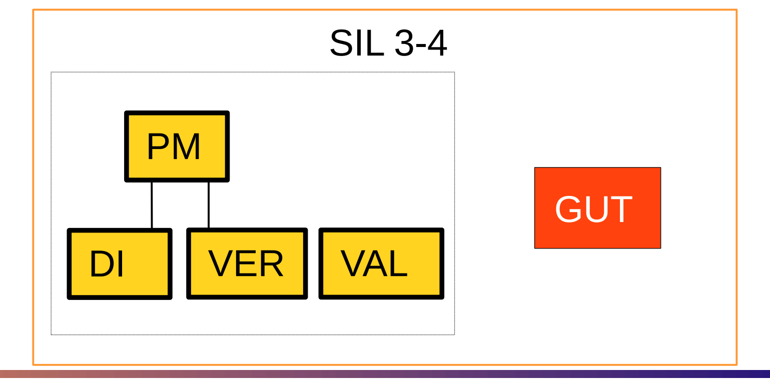
(dünner Rahmen) darf dieselbe Organisation sein

(dicker Rahmen) darf dieselbe Person sein









ASIL nach ISO 26262 (1/2)

– SIL-Stufen aus der Grundnorm DIN 61508 können direkt angewandt oder für spezifische Bereiche angepasst werden

– In der Norm ISO 26262 wurde SIL in den Automotive-Bereich überführt und mit ASIL (*Automotive Safety Integrity Levels*) bezeichnet

– ASIL ist ein qualitatives Maß für die Sicherheitsrelevanz einer Fehlfunktion und ergibt sich aus den Parametern:

- Auftretenswahrscheinlichkeit E (*Exposure*)

  Wie häufig sind Situationen, in denen die Fehlfunktion relevant ist?

- Kontrollierbarkeit C (*Controllability*)

  Wenn die Fehlfunktion auftritt, wie gut kann sie beherrscht werden?

- Schwere der Schädigung S (*Severity*)

  Wenn die Fehlfunktion nicht beherrscht wird, wie groß ist die Auswirkung?

ASIL nach ISO 26262 (2/2)

- Methodik der ASIL Einstufung:
  - Skala A bis D, wobei A die niedrigste und D die höchste Einstufung einer Fehlfunktion darstellt
  - Einstufung als QM (Qualitätsmanagement), d.h., dass eine Fehlfunktion nicht als sicherheitsrelevant eingestuft wird
- Beispiele für ASIL von Gefährdungen aus der Praxis:
  - Airbag löst ohne Bedarfsfall aus
  - Ungewollte Beschleunigung
  - Lenkunterstützung fällt aus
- Nach der Einstufung einer Fehfunktion erfolgt die Ableitung von Maßnahmen, falls eine Reduzierung der Stufe nötig ist / möglich ist

Frage zu Kapitel 6.4

Welchen Aussagen stimmen Sie zu?

☐ Der Tod mehrerer Personen ist nicht über eine SIL-Stufe erfassbar und muss bei jedem System ausgeschlossen werden.

☐ SIL repräsentiert ein quantitatives Verfahren zur Risikobewertung.

☐ Nach der Unabhängigkeit nach SIL 0 kann der Designer auch seine Arbeit validieren.

☐ Nach der Unabhängigkeit nach SIL 2-3 sind Verifikation und Validation dem gleichen Projektmanagement unterworfen wie das Design.

Gliederung

§7 Übungsaufgaben

Aufgabe 1 – Zuverlässigkeitsblockdiagramm

Aufgabe 2 – Fehlerbaumanalyse (FTA)

Aufgabe 3 – Softwarezuverlässigkeit

Aufgabe 1 - Aufgabenstellung

Ein Steuergerät wird aus einem Mikrocontrollersystem und einem Spannungs-Versorgungssystem aufgebaut.

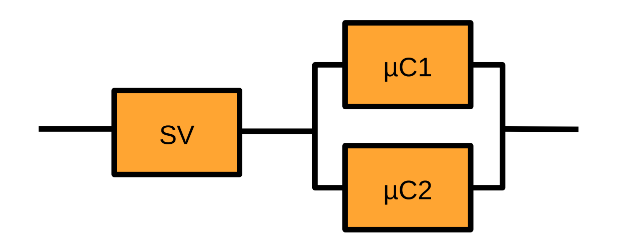
Für die Ausfallraten der Teilsystem gilt: $\lambda_{\mu C} = 8 \cdot 10^{-6} h^{-1}$ $\lambda_{SV} = 3 \cdot 10^{-5} h^{-1}$

A) Die Zuverlässigkeit des Gesamtsystems muss erhöht werden, daher sollen die einzelnen Teilsysteme jeweils redundant ausgeführt werden. Ist es durch einfache Redundanz möglich, eine Zuverlässigkeit von 75% bei einer Betriebszeit von 1,5 Jahren zu erreichen?

B) Wenn nein, wie ist dies durch Redundanz zu erreichen? Aus wirtschaftlichen Gründen darf nur ein Teilsystem maximal aus 3 Komponenten ausgeführt werden.

C) Welche Verfügbarkeit weist das System für die höchstmögliche Zuverlässigkeit auf, wenn eine Reparatur ca. 3 Tage in Anspruch nimmt?

Lösung Teil A (1/4)



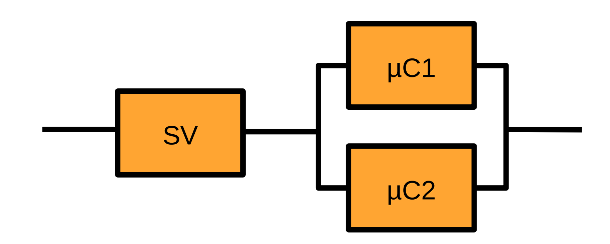
Möglichkeit 1: Mikrocontroller redundant

Für die einzelnen Komponenten gilt:

Parallelschaltung von µC1 und µC2:

Serienschaltung von SV und µC1/µC2:

Lösung Teil A (2/4)



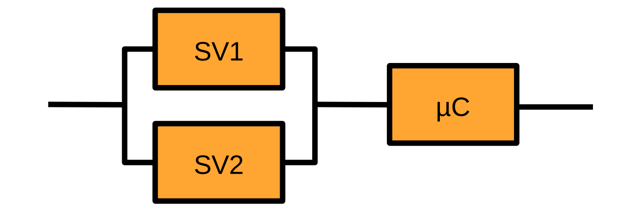
Möglichkeit 1: Mikrocontroller redundant

Für die gesamte Ausfallrate gilt:

Für die Zuverlässigkeit folgt:

Lösung Teil A (3/4)

Möglichkeit 2: Spannungsversorgung redundant

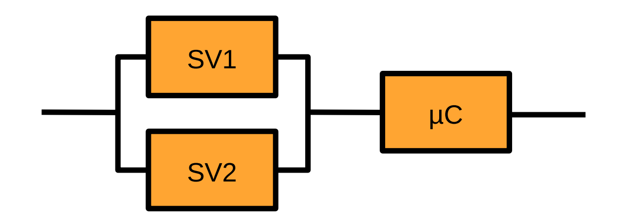


Für die einzelnen Komponenten gilt:

Parallelschaltung von SV1 und SV2:

Serienschaltung von SV und µC1/µC2:

Lösung Teil A (4/4)



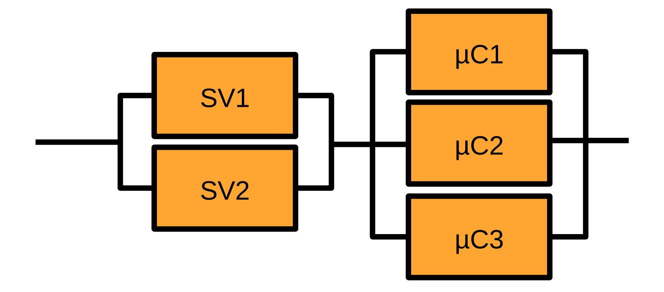
Möglichkeit 2: Spannungsversorgung redundant

Für die gesamte Ausfallrate gilt:

Für die Zuverlässigkeit folgt:

Lösung Teil B (1/4)

Möglichkeit 3: Mehrfache Redundanz nach dem Schema:



Es gilt:

Parallelschaltung von SV1 und SV2:

Parallelschaltung von µC1/µC2/µC3:

Lösung Teil B (2/4)

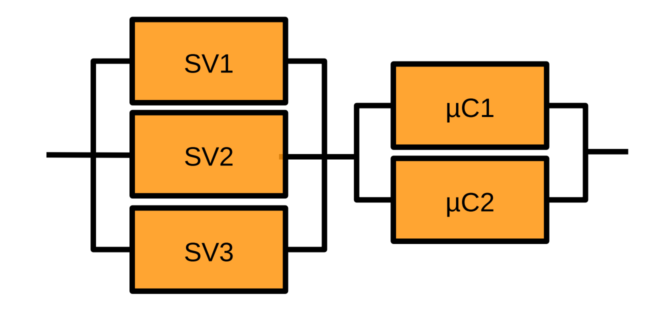
Möglichkeit 3: Mehrfache Redundanz nach dem Schema:

Serienschaltung von SV1/SV2 und µC1/µC2/µC3:

Ausfallrate und Zuverlässigkeit:

Lösung Teil B (3/4)

Möglichkeit 4: Mehrfache Redundanz nach dem Schema:

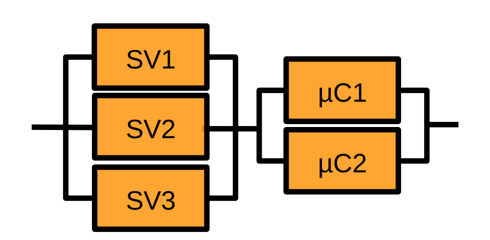


Es gilt:

Parallelschaltung von SV1/SV2/SV3:

Parallelschaltung von µC1 und µC2 :

Lösung Teil B (4/4)



Möglichkeit 3: Mehrfache Redundanz nach dem Schema:

Serienschaltung von SV1/SV2/SV3 und µC1/µC2:

Ausfallrate und Zuverlässigkeit:

Lösung Teil C

Für die Verfügbarkeit des Systems folgt:

- Berechnung der MTTR:

- Berechnung der Verfügbarkeit:

§7 Übungsaufgaben

Aufgabe 1 – Zuverlässigkeitsblockdiagramm

Aufgabe 2 – Fehlerbaumanalyse (FTA)

Aufgabe 3 – Softwarezuverlässigkeit

Aufgabe 2 - Aufgabenstellung

Ein System realisiert die Regelung des Kompressor-Drucks für eine verfahrenstechnische Anlage. Um die geforderte Funktion zu erfüllen, benötigt die Anlage einen Minimaldruck von 4 bar. Die Anlage soll 12 Jahre im Betrieb sein.

Das Automatisierungssystem besteht aus einem Drucksensor, einer Druckluftpumpe, der Spannungsversorgung für die Pumpe, einem Regelsystem (Mikrocontrollersteuerung) und einer Spannungsversorgung für die Mikrocontrollersteuerung. Ein zu hoher Kompressor-Druck soll in dieser Betrachtung nicht berücksichtigt werden, dieser soll über ein Überdruckventil verhindert werden.

Führen Sie für dieses System eine quantitative Fehlerbaumanalyse durch.

Aufgabe 2 - Kenngrößen

Die Ausfallraten sind wie folgt gegeben:

$$\lambda_{\text{Pumpe}} = 4 \cdot 10^{-9} h^{-1}$$

$$\lambda_{\text{Spg.-Pumpe}} = 2 \cdot 10^{-7} h^{-1}$$

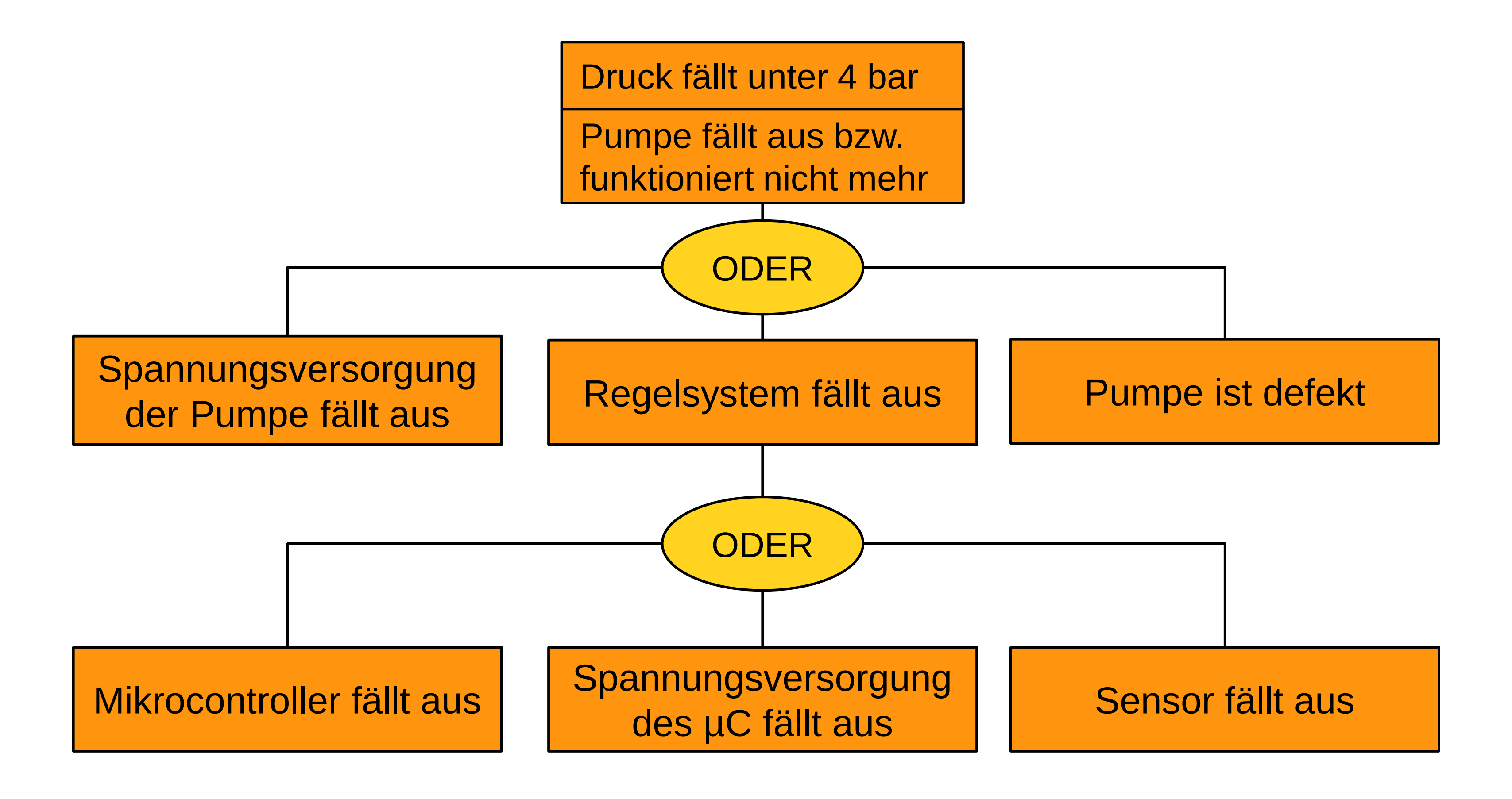
$$\lambda_{\mu C} = 8 \cdot 10^{-8} h^{-1}$$

$$\lambda_{\text{Spg.-}\mu C} = 2 \cdot 10^{-7} h^{-1}$$

$$\lambda_{\text{Sensor}} = 3 \cdot 10^{-9} h^{-1}$$

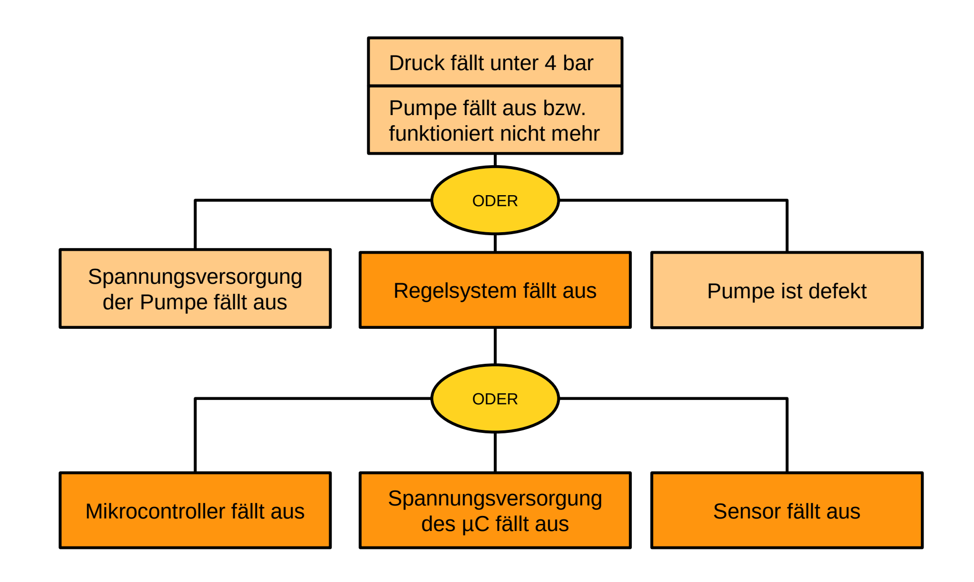
Für die Betriebszeit gilt:

Lösung Schritt 1 – Erstellen des Fehlerbaums



Lösung Schritt 2 – Unterbaum Zusammenfassen

ODER-Verknüpfung wird nach Schema wie folgt zusammengefasst (vgl. Kap 3.3):

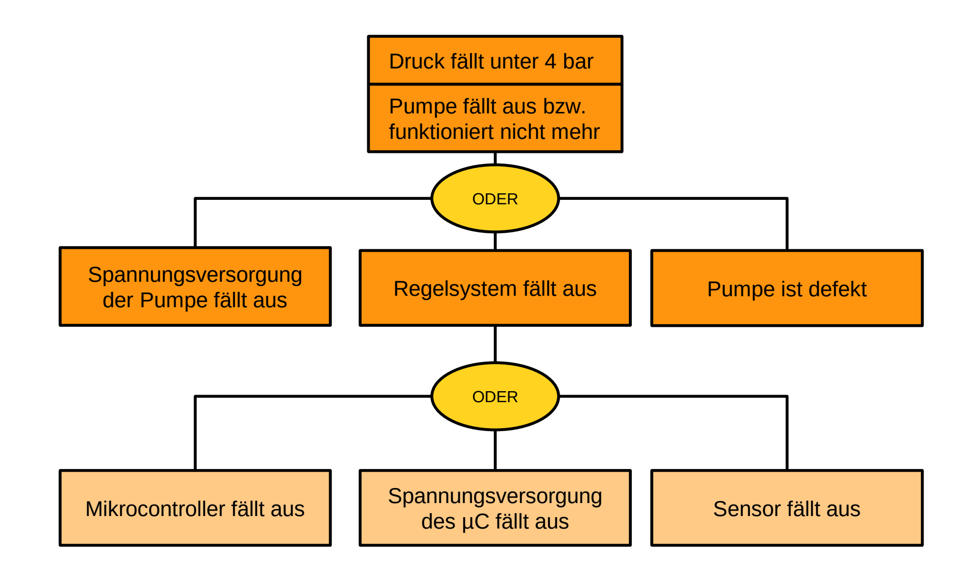


Lösung Schritt 3– Gesamtzuverlässigkeit berechnen

ODER-Verknüpfung wird nach

Schema wie folgt zusammengefasst

(vgl. Kap 3.3):



§7 Übungsaufgaben

Aufgabe 1 – Zuverlässigkeitsblockdiagramm

Aufgabe 2 – Fehlerbaumanalyse (FTA)

Aufgabe 3 – Softwarezuverlässigkeit

Aufgabe 3 - Aufgabenstellung

Ein Software-Programm mit 5000 Anweisungen wurde 25 Tage getestet.

Durchschnittlich lief das Programm 6 Stunden pro Tag.

Es wurden dabei 20 Fehler gefunden und korrigiert.

Die Proportionalitätskonstante beträgt bei diesem Projekt 8 h^{-1}.

Berechnen Sie die Fehlerrate und die Gesamtzahl der sich im Programm befindlichen Fehler.

Lösung (1/2)

Es gilt:

- Berechnung der Fehlerrate:

Lösung (2/2)

Es gilt:

- Berechnung der Fehlerzahl:

Literatur

[Göhn12a] **Göhner, Peter**: Skript zur Vorlesung Automatisierungstechnik II, IAS, Stuttgart 2012.

[Göhn12b] **Göhner, Peter**: Skript zur Vorlesung Softwaretechnik I, IAS, Stuttgart 2012.

[Göhn12c] **Göhner, Peter**: Skript zur Vorlesung Softwaretechnik II, IAS, Stuttgart 2012.

[Jazd14] **Jazdi, Nasser**: Skript zur Vorlesung Zuverlässigkeit und Sicherheit von Automatisierungssystemen, IAS, Stuttgart, 2014.

[Bles11] **Blessing, Peter**: Skript zur Vorlesung Sicherheit und Zuverlässigkeit, HSHN, Heilbronn, 2011.

[Bert10] **Bertsche, Bernd**: Skript zur Vorlesung Zuverlässigkeitstechnik I und II, IMA, Stuttgart, 2010.

[BGJ+09] **Bertsche, Bernd; Göhner, Peter; Jensen, Uwe; Schinköthe, Wolfgang; Wunderlich, Hans-Joachim:** Zuverlässigkeit mechatronischer Systeme, 1.Aufl., Berlin, Heidelberg, Springer-Verlag, 2009**.**

[MePa10] **Meyna, Arno; Pauli, Bernhard**: Zuverlässigkeitstechnik – Quantitative Bewertungsverfahren, 2.Aufl., München, Wien, Hanser-Verlag, 2010.

[Balz08] **Balzert, Helmut**: Lehrbuch der Softwaretechnik, 2.Aufl., Heidelberg, Spektrum Akademischer Verlag, 2008.

[BeHa09] **Benra, Juliane; Halang, Wolfgang**: Software-Entwicklung für Echtzeitsysteme, 1.Aufl., Berlin, Heidelberg, Springer-Verlag, 2009.

[Ligg09] **Liggesmeyer, Peter**: Software Qualität -Testen, Analysieren und Verifizieren von Software, 2.Auflg., Heidelberg, Spektrum Akademischer Verlag, 2009

[Hala13] **Halang, Wolfgang**: Funktionale Sicherheit – Echtzeit 2013, 1.Auflg., Berlin, Heidelberg, Springer-Verlag, 2013